ASC X12C/005010X230

Based on Version 5, Release 1

ASC X12 Standards for Electronic Data Interchange
Technical Report Type 3

Functional Acknowledgment (997)

APRIL 2005

$67.50 - Bound Document
$45.00 - Downloaded Portable Document (PDF)
$80.00 - PDF on CD-ROM

Manager, Publications and Standards
Data Interchange Standards Association, Inc.
7600 Leesburg Pike • Suite 430 • Falls Church, VA 22043
Phone: (703) 548-7005
Email: publications@disa.org

Table of Contents

1 Purpose and Business Information

1.1 Implementation Purpose and Scope

The purpose of this implementation guide is to provide standardized data content and structure to users of the ASC X12 997 transaction set. This implementation guide is intended to enable a receiver of an X12 transaction set to report syntactical errors against the X12 standard, or to acknowledge receipt of an error-free transaction set.

This 997 implementation guide can NOT be used for responding to any implementation guideline (TR3).

This 997 implementation guide can NOT be used for any application level validations.

1.2 Version Information

This implementation guide is based on the October 2003 ASC X12 standards, referred to as Version 5, Release 1, Sub-release 0 (005010). The unique Version/Release/Industry Identifier Code for transaction sets that are defined by this implementation guide is 005010**X230**.

The two-character Functional Identifier Code(s) for the transaction set(s) that are included in this implementation guide is (are):

- ***FA*** ***Functional or Implementation Acknowledgment Transaction Sets (997, 999)***

The Version/Release/Industry Identifier Code and the applicable Functional Identifier Code must be transmitted in the Functional Group Header (GS segment) that begins a functional group of these transaction sets. For more information, see the descriptions of GS01 and GS08 in Appendix C.

1.3 Implementation Limitations

1.3.1 Batch and Real-Time Usage

There are multiple methods available for sending and receiving business transactions electronically. Two common modes for EDI transactions are batch and real-time.

Batch - In a batch mode the sender does not remain connected while the receiver processes the transactions. Processing is usually completed according to a set schedule. If there is an associated business response transaction (such as a 271 Response to a 270 Request for Eligibility), the receiver creates the response transaction and stores it for future delivery. The sender of the original transmission reconnects at a later time and picks up the response transaction. This implementation guide does not set specific response time parameters for these activities.

Real Time - In real-time mode the sender remains connected while the receiver processes the transactions and returns a response transaction to the sender. This implementation guide does not set specific response time parameters for implementers.

This implementation guide is intended to support use in batch mode. This implementation guide is intended to support use in real-time mode.

1.3.2 Other Usage Limitations

The ASC X12 997 transaction set is designed to report only on syntactical conformance against the X12 standard.

This 997 implementation guide can NOT be used for responding to any implementation guideline (TR3).

This 997 implementation guide can NOT be used for any application level validations.

The ASC X12 997 transaction set is designed to respond to one and only one functional group (i.e. GS/GE), but may respond to multiple transaction sets (i.e. ST/SE) within that functional group.

This ASC X12 997 Implementation Guideline can NOT be used to respond to any management transactions sets intended for acknowledgements, i.e., TS 997 and 999, or interchange control segments related to acknowledgements, i.e. TA1 and TA3.

1.4 Business Usage

This ASC X12 997 implementation guide is intended to meet the needs of the EDI industry as a whole, for a standard implementation guideline designed for reporting of syntactical errors against the X12 standard, or to report receipt of a transaction set that fully complies with the X12 standard.

For more information on the relationship between the 997 transaction set and other response transaction sets, refer to the ASC X12 document “Reference Model for the Acknowledgement and Tracking of EDI Interchanges”.

1.5 Business Terminology

No special business terms are used in this implementation guide.

1.6 Transaction Acknowledgments

There are several acknowledgement transactions available for use. The recommendations of the implementation guide authors are noted in this section. Other acknowledgement transactions may be used at the discretion of the trading partners.

1.6.1 997 Functional Acknowledgment

The 997 informs the submitter that the functional group arrived at the destination. It may include information about the syntactical quality of the functional group.

The Functional Acknowledgment (997) transaction is not required as a response to receipt of a batch transaction compliant with this implementation guide.

The Functional Acknowledgment (997) transaction is not required as a response to receipt of a real-time transaction compliant with this implementation guide.

1.6.2 999 Implementation Acknowledgment

The 999 informs the submitter that the functional group arrived at the destination. It may include information about the syntactical quality of the functional group and the implementation guide compliance.

The Functional Acknowledgment (999) transaction is not required as a response to receipt of a batch transaction compliant with this implementation guide.

The Functional Acknowledgment (999) transaction is not required as a response to receipt of a real-time transaction compliant with this implementation guide.

1.6.3 824 Application Advice

The 824 informs the submitter that the transaction set arrived at the destination. It may include information about the syntactical quality of the transaction set and the implementation guide compliance.

The Application Advice (824) transaction is not required as a response to receipt of a batch transaction compliant with this implementation guide.

The Application Advice (824) transaction is not required as a response to receipt of a real-time transaction compliant with this implementation guide.

1.7 Related Transactions

There are no transactions related to the transactions described in this implementation guide.

1.8 Trading Partner Agreements

Trading partner agreements are used to establish and document the relationship between trading partners. A trading partner agreement must not override the specifications in this implementation guide if a transmission is reported in GS08 to be a product of this implementation guide

1.9 Data Overview

1.9.1 Overall Data Architecture

NOTE
See Appendix B, Nomenclature, to review the transaction set structure, including descriptions of segments, data elements, levels and loops.

1.9.1.1 Response Process

The following informational flow shows how the 997 transaction set is used with other X12 response transactions.

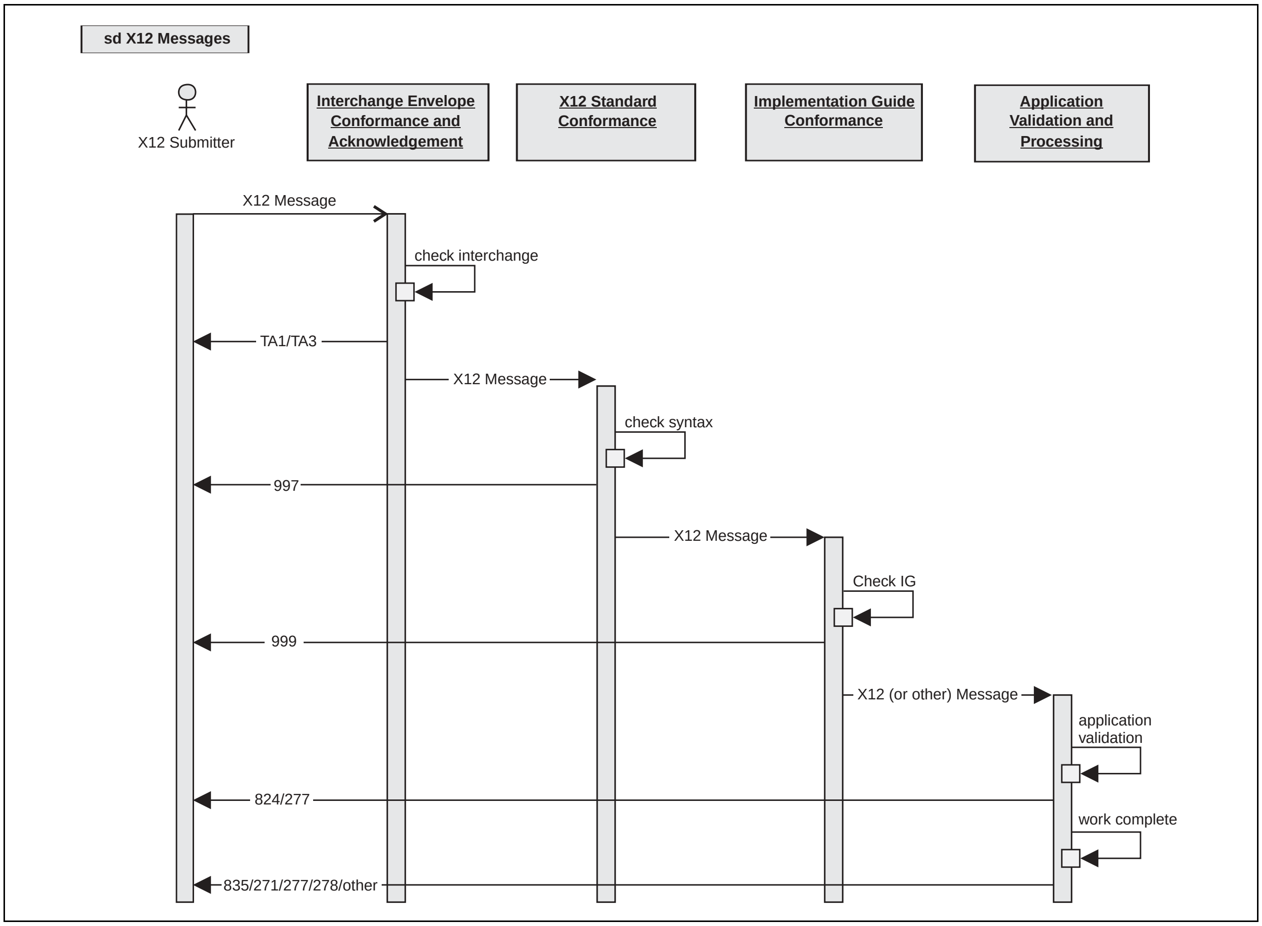


Figure 1.1. Information Flow

For more information on the relationship between the 997 transaction set and other response transaction sets, refer to the ASC X12 document “Reference Model for the Acknowledgement and Tracking of EDI Interchanges”.

2 Transaction Set

NOTE
See Appendix B, Nomenclature, to review the transaction set structure, including descriptions of segments, data elements, levels, and loops.

2.1 Presentation Examples

The ASC X12 standards are generic. For example, multiple trading communities use the same PER segment to specify administrative communication contacts. Each community decides which elements to use and which code values in those elements are applicable.

In this implementation guide, **IMPLEMENTATION** specifies the requirements for this implementation. **X12 STANDARD** is included as a reference only.

The transaction set presentation is comprised of two main sections with subsections within the main sections:

2.3 Transaction Set Listing

There are two sub-sections under this general title. The first sub-section concerns this implementation of a generic X12 transaction set. The second sub-section concerns the generic X12 standard itself.

IMPLEMENTATION

This section lists the levels, loops, and segments contained in this implementation. It also serves as an index to the segment detail.

STANDARD

This section is included as a reference.

2.4 Segment Detail

There are three sub-sections under this general title. This section repeats once for each segment used in this implementation providing segment specific detail and X12 standard detail.

SEGMENT DETAIL

This section is included as a reference.

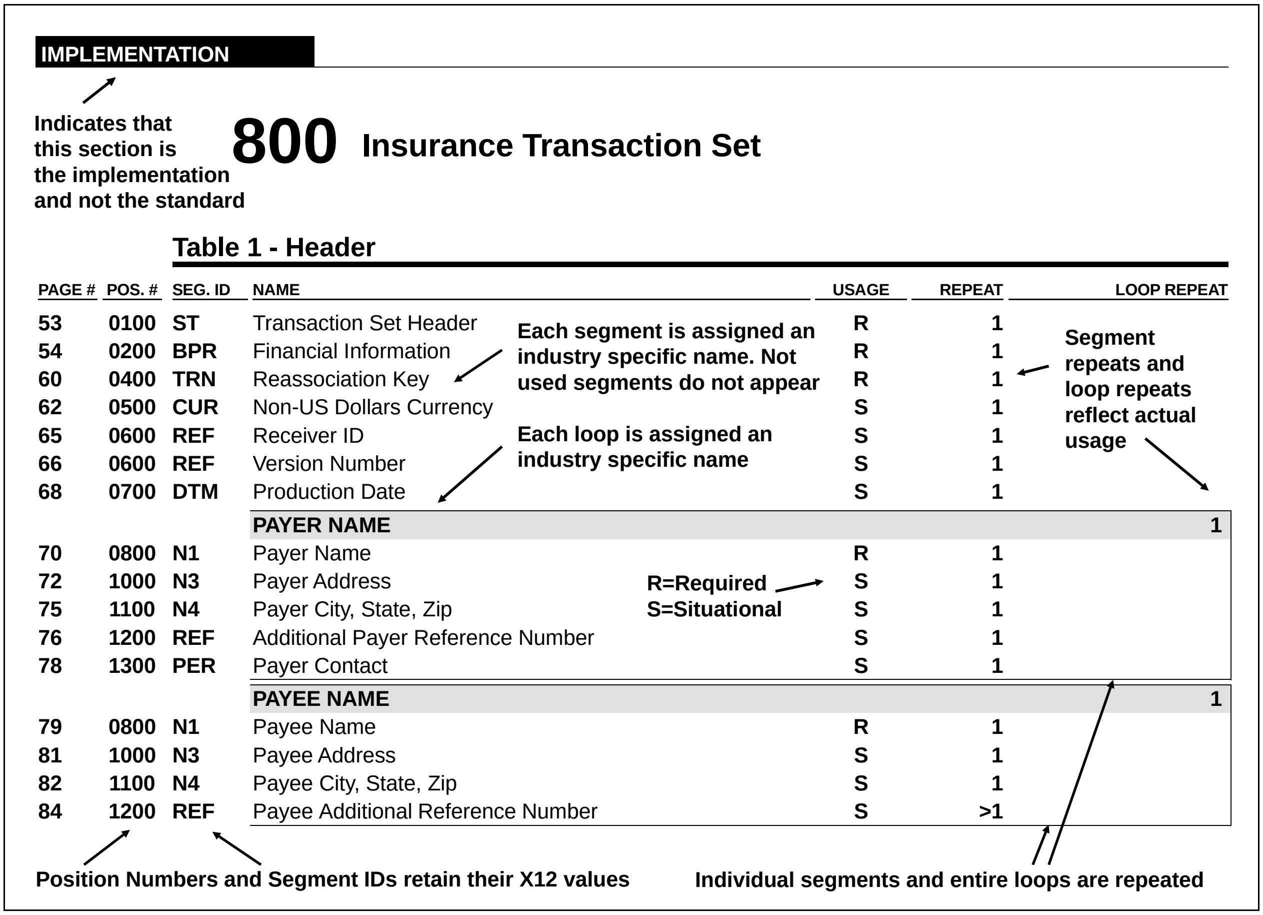
DIAGRAM

This section is included as a reference. It provides a pictorial view of the standard and shows which elements are used in this implementation.

ELEMENT DETAIL

This section specifies the implementation details of each data element.

These illustrations (Figures 2.1 through 2.5) are examples and are not extracted from the Section 2 detail in this implementation guide. Annotated illustrations, presented below in the same order they appear in this implementation guide, describe the format of the transaction set that follows.



IMPLEMENTATION

Indicates that this section is the implementation and not the standard

800 Insurance Transaction Set

Table 1 - Header

PAGE #	POS. #	SEG. ID	NAME	USAGE	REPEAT	LOOP REPEAT
53	0100	ST	Transaction Set Header	R	1	
54	0200	BPR	Financial Information	R	1	
60	0400	TRN	Reassociation Key	R	1	
62	0500	CUR	Non-US Dollars Currency	S	1	
65	0600	REF	Receiver ID	S	1	
66	0600	REF	Version Number	S	1	
68	0700	DTM	Production Date	S	1	
			PAYER NAME			1
70	0800	N1	Payer Name	R	1	
72	1000	N3	Payer Address	S	1	
75	1100	N4	Payer City, State, Zip	S	1	
76	1200	REF	Additional Payer Reference Number	S	1	
78	1300	PER	Payer Contact	S	1	
			PAYEE NAME			1
79	0800	N1	Payee Name	R	1	
81	1000	N3	Payee Address	S	1	
82	1100	N4	Payee City, State, Zip	S	1	
84	1200	REF	Payee Additional Reference Number	S	>1	

Each segment is assigned an industry specific name. Not used segments do not appear

Each loop is assigned an industry specific name

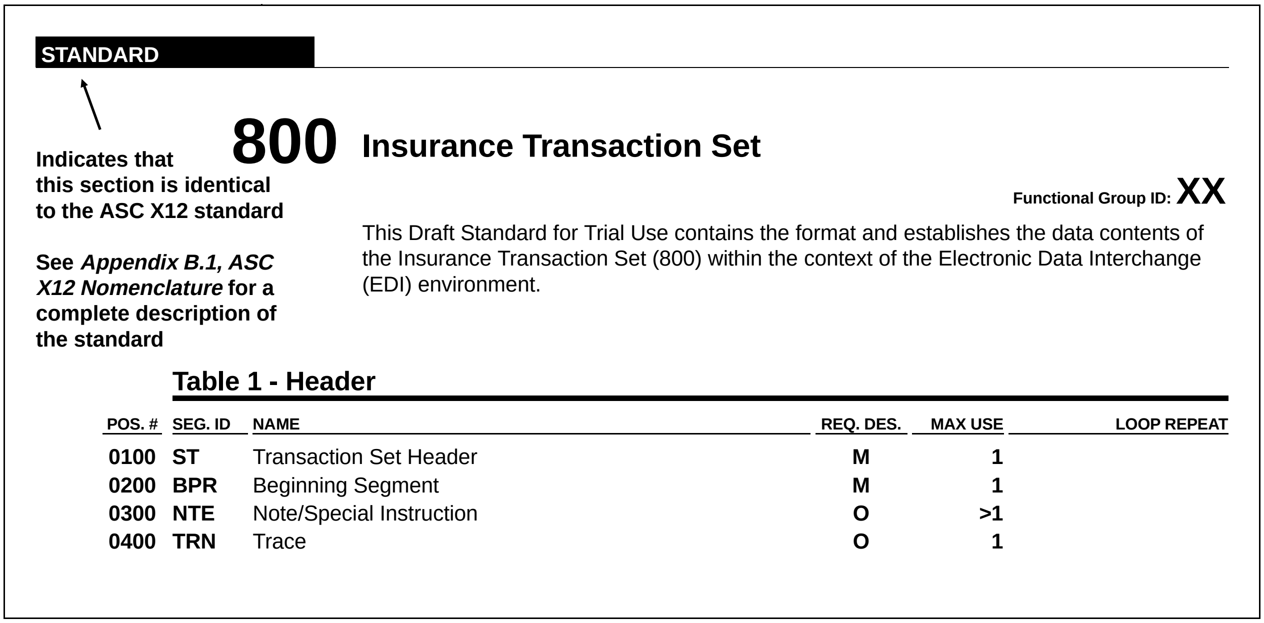
Segment repeats and loop repeats reflect actual usage

R=Required
S=Situational

Position Numbers and Segment IDs retain their X12 values

Individual segments and entire loops are repeated

Figure 2.1. Transaction Set Key — Implementation



STANDARD

Indicates that this section is identical to the ASC X12 standard

See *Appendix B.1, ASC X12 Nomenclature* for a complete description of the standard

800 Insurance Transaction Set

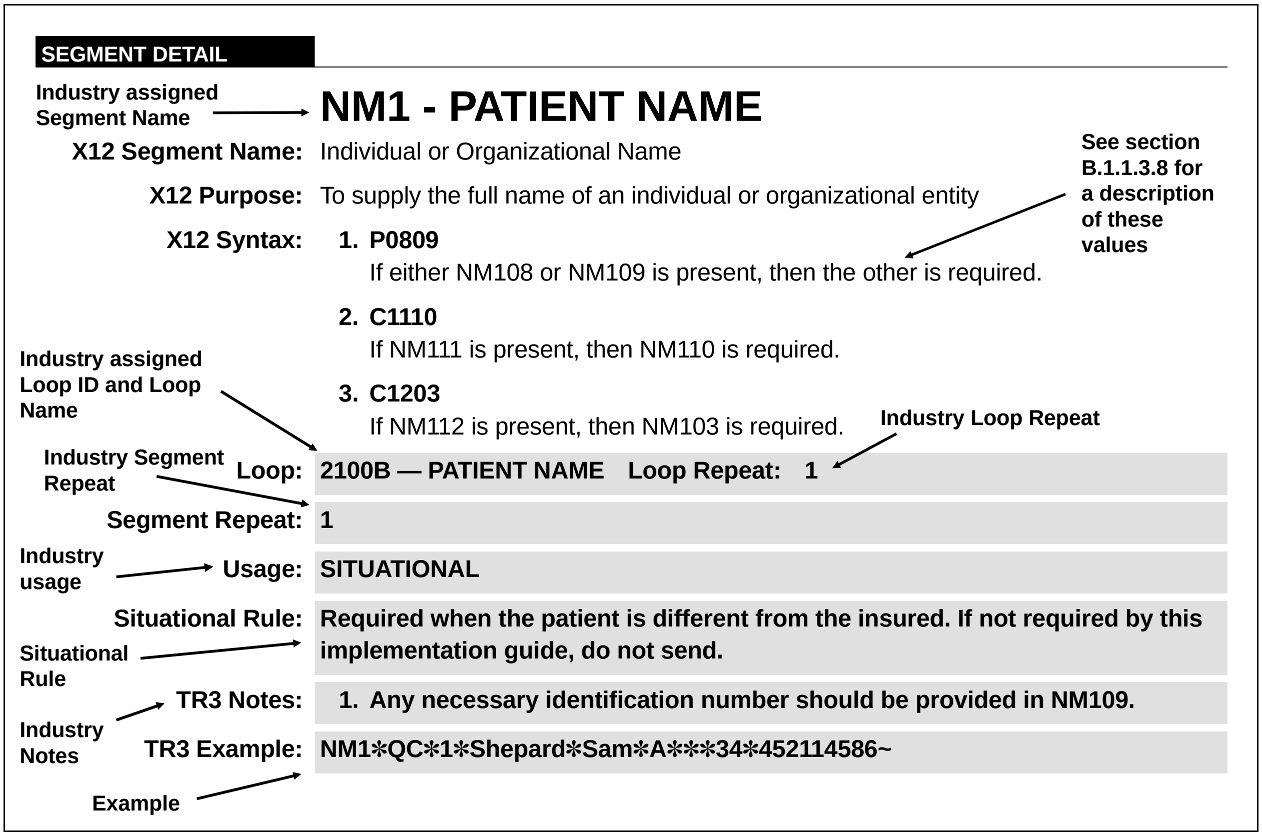
Functional Group ID: **XX**

This Draft Standard for Trial Use contains the format and establishes the data contents of the Insurance Transaction Set (800) within the context of the Electronic Data Interchange (EDI) environment.

Table 1 - Header

POS. #	SEG. ID	NAME	REQ. DES.	MAX USE	LOOP REPEAT
0100	ST	Transaction Set Header	M	1	
0200	BPR	Beginning Segment	M	1	
0300	NTE	Note/Special Instruction	O	>1	
0400	TRN	Trace	O	1	

Figure 2.2. Transaction Set Key — Standard



SEGMENT DETAIL

NM1 - PATIENT NAME

X12 Segment Name: Individual or Organizational Name

X12 Purpose: To supply the full name of an individual or organizational entity

X12 Syntax:

1. **P0809**
   If either NM108 or NM109 is present, then the other is required.
2. **C1110**
   If NM111 is present, then NM110 is required.
3. **C1203**
   If NM112 is present, then NM103 is required.

Loop: **2100B — PATIENT NAME Loop Repeat: 1**

Segment Repeat: **1**

Usage: **SITUATIONAL**

Situational Rule: **Required when the patient is different from the insured. If not required by this implementation guide, do not send.**

TR3 Notes: **1. Any necessary identification number should be provided in NM109.**

TR3 Example: **NM1✻QC✻1✻Shepard✻Sam✻A✻✻✻34✻452114586~**

Industry assigned Segment Name
See section B.1.1.3.8 for a description of these values
Industry assigned Loop ID and Loop Name
Industry Loop Repeat
Industry Segment Repeat
Industry usage
Situational Rule
Industry Notes
Example

Figure 2.3. Segment Key — Implementation

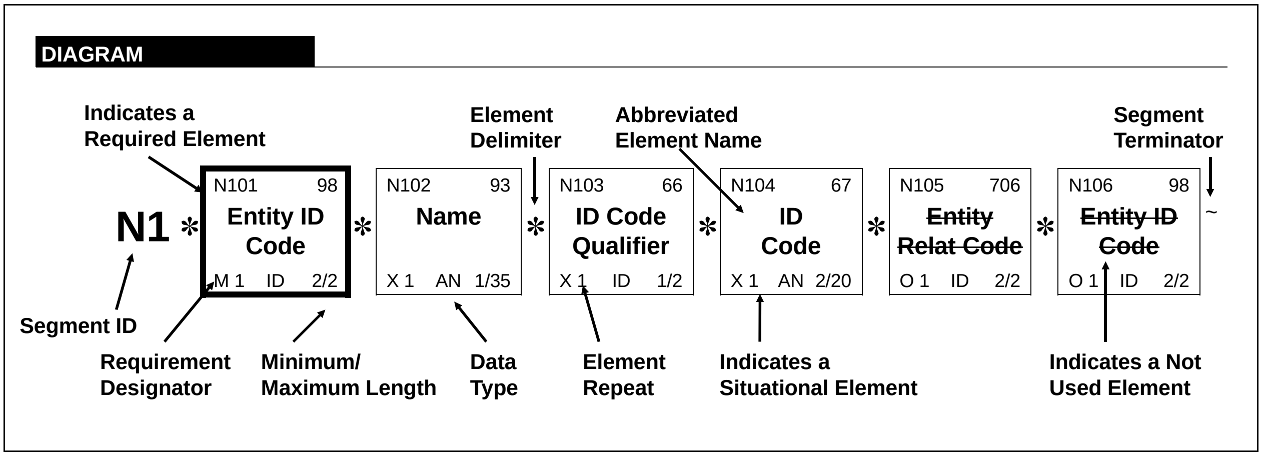
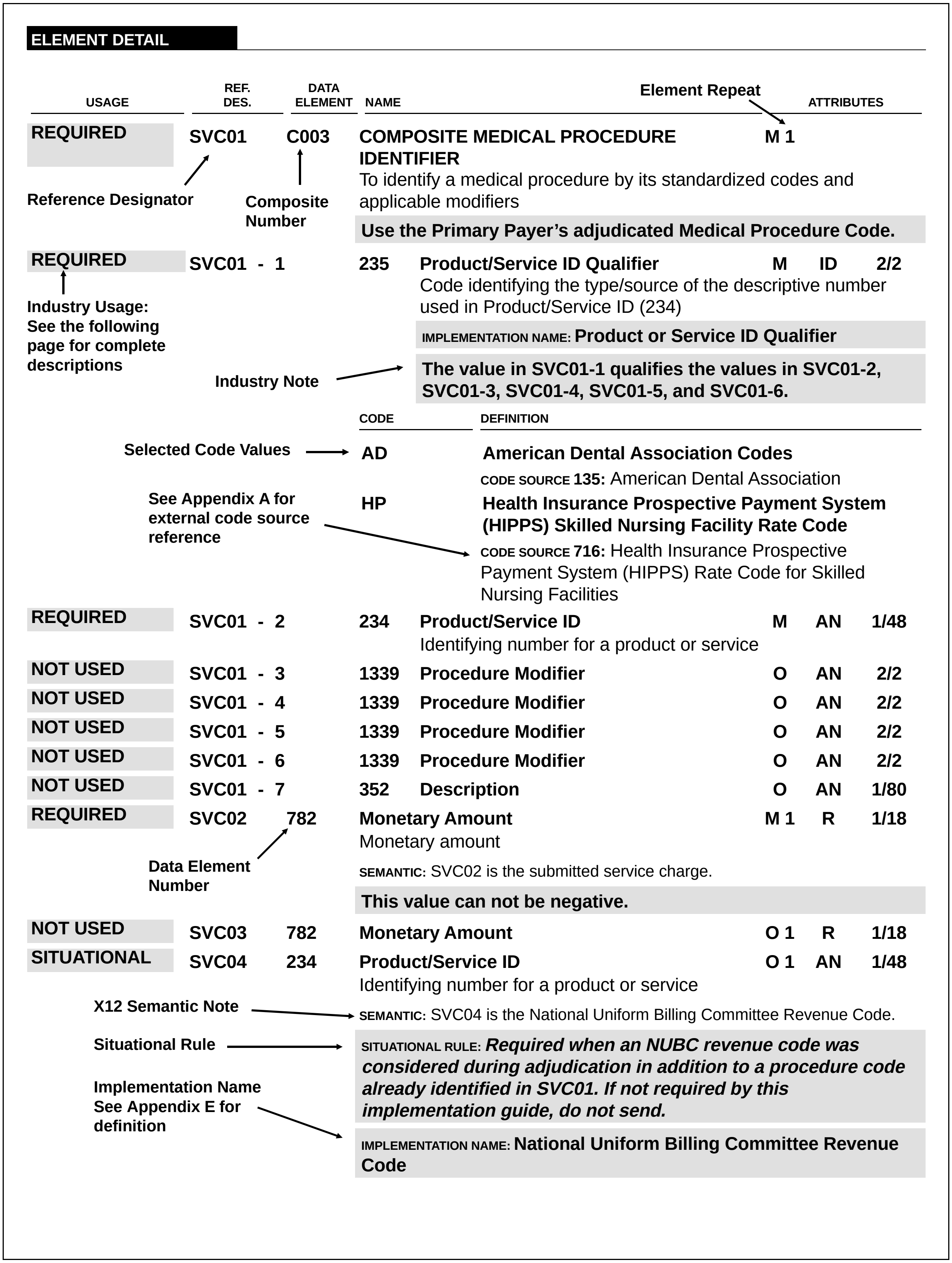


Figure 2.4. Segment Key — Diagram



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	SVC01	C003	**COMPOSITE MEDICAL PROCEDURE IDENTIFIER** To identify a medical procedure by its standardized codes and applicable modifiers **Use the Primary Payer’s adjudicated Medical Procedure Code.**	M 1		
REQUIRED	SVC01 - 1	235	**Product/Service ID Qualifier** Code identifying the type/source of the descriptive number used in Product/Service ID (234) IMPLEMENTATION NAME: **Product or Service ID Qualifier** **The value in SVC01-1 qualifies the values in SVC01-2, SVC01-3, SVC01-4, SVC01-5, and SVC01-6.** CODE DEFINITION **AD** **American Dental Association Codes** CODE SOURCE **135:** American Dental Association **HP** **Health Insurance Prospective Payment System (HIPPS) Skilled Nursing Facility Rate Code** CODE SOURCE **716:** Health Insurance Prospective Payment System (HIPPS) Rate Code for Skilled Nursing Facilities	M	ID	2/2
REQUIRED	SVC01 - 2	234	**Product/Service ID** Identifying number for a product or service	M	AN	1/48
NOT USED	SVC01 - 3	1339	**Procedure Modifier**	O	AN	2/2
NOT USED	SVC01 - 4	1339	**Procedure Modifier**	O	AN	2/2
NOT USED	SVC01 - 5	1339	**Procedure Modifier**	O	AN	2/2
NOT USED	SVC01 - 6	1339	**Procedure Modifier**	O	AN	2/2
NOT USED	SVC01 - 7	352	**Description**	O	AN	1/80
REQUIRED	SVC02	782	**Monetary Amount** Monetary amount SEMANTIC: SVC02 is the submitted service charge. **This value can not be negative.**	M 1	R	1/18
NOT USED	SVC03	782	**Monetary Amount**	O 1	R	1/18
SITUATIONAL	SVC04	234	**Product/Service ID** Identifying number for a product or service SEMANTIC: SVC04 is the National Uniform Billing Committee Revenue Code. SITUATIONAL RULE: ***Required when an NUBC revenue code was considered during adjudication in addition to a procedure code already identified in SVC01. If not required by this implementation guide, do not send.*** IMPLEMENTATION NAME: **National Uniform Billing Committee Revenue Code**	O 1	AN	1/48

Element Repeat
Reference Designator
Composite Number
Industry Usage: See the following page for complete descriptions
Industry Note
Selected Code Values
See Appendix A for external code source reference
Data Element Number
X12 Semantic Note
Situational Rule
Implementation Name See Appendix E for definition

Figure 2.5. Segment Key — Element Summary

2.2 Implementation Usage

2.2.1 Industry Usage

Industry Usage describes when loops, segments, and elements are to be sent when complying with this implementation guide. The three choices for Usage are required, not used, and situational. To avoid confusion, these are named differently than the X12 standard Condition Designators (mandatory, optional, and relational).

Required This loop/segment/element must always be sent.

Required segments in Situational loops only occur when the loop is used.

Required elements in Situational segments only occur when the segment is used.

Required component elements in Situational composite elements only occur when the composite element is used.

Not Used This element must never be sent.

Situational Use of this loop/segment/element varies, depending on data content and business context as described in the defining rule. The defining rule is documented in a Situational Rule attached to the item.

There are two forms of Situational Rules.

The first form is “Required when <explicit condition statement>. If not required by this implementation guide, may be provided at the sender’s discretion, but cannot be required by the receiver.” The data qualified by such a situational rule cannot be required or requested by the receiver, transmission of this data is solely at the sender’s discretion.

The alternative form is “Required when <explicit condition statement>. If not required by this implementation guide, do not send.” The data qualified by such a situational rule cannot be sent except as described in the explicit condition statement.

2.2.2 Transaction Compliance Related to Industry Usage

A transmitted transaction complies with an implementation guide when it satisfies the requirements as defined within the implementation guide. The presence or absence of an item (loop, segment, or element) complies with the industry usage specified by this implementation guide according to the following table.

Industry Usage	Business Condition is	Item is	Transaction Complies with Implementation Guide?
Required	N/A	Sent	Yes
		Not Sent	No
Not Used	N/A	Sent	No
		Not Sent	Yes
Situational (Required when <explicit condition statement>. If not required by this implementation guide, may be provided at the sender’s discretion, but cannot be required by the receiver.)	True	Sent	Yes
		Not Sent	No
	Not True	Sent	Yes
		Not Sent	Yes
Situational (Required when <explicit condition statement>. If not required by this implementation guide, do not send.)	True	Sent	Yes
		Not Sent	No
	Not True	Sent	No
		Not Sent	Yes

This table specifies how an entity is to evaluate a transmitted transaction for compliance with industry usage. It is not intended to require or imply that the receiver must reject non-compliant transactions. The receiver will handle non-compliant transactions based on its business process and any applicable regulations.

2.2.3 Loops

Loop requirements depend on the context or location of the loop within the transaction. See Appendix B for more information on loops.

- A nested loop can be used only when the associated higher level loop is used.
- The usage of a loop is the same as the usage of its beginning segment.
    - If a loop’s beginning segment is Required, the loop is Required and must occur at least once unless it is nested in a loop that is not being used.
    - If a loop’s beginning segment is Situational, the loop is Situational.
- Subsequent segments within a loop can be sent only when the beginning segment is used.
- Required segments in Situational loops occur only when the loop is used.

2.3 Transaction Set Listing

2.3.1 Implementation

This section lists the levels, loops, and segments contained in this implementation. It also serves as an index to the segment detail. Refer to section 2.1 Presentation Examples for detailed information on the components of the Implementation section.

IMPLEMENTATION

997 Functional Acknowledgment

Table 1 - Header

PAGE #	POS. #	SEG. ID	NAME	USAGE	REPEAT	LOOP REPEAT
20	0100	ST	Transaction Set Header	R	1	
22	0200	AK1	Functional Group Response Header	R	1	
			LOOP ID - 2000 AK2 TRANSACTION SET RESPONSE			999999
24	0300	AK2	AK2 Transaction Set Response	S	1	
			LOOP ID - 2100 AK2/AK3 ERROR IDENTIFICATION			999999
26	0400	AK3	AK2/AK3 Error Identification	S	1	
28	0500	AK4	Data Element Note	S	99	
30	0600	AK5	Transaction Set Response	R	1	
35	0700	AK9	Functional Group Response Trailer	R	1	
40	0800	SE	Transaction Set Trailer	R	1	

2.3.2 X12 Standard

This section is included as a reference. The implementation guide reference clarifies actual usage. Refer to section 2.1 Presentation Examples for detailed information on the components of the X12 Standard section.

STANDARD

997 Functional Acknowledgment

Functional Group ID: FA

This X12 Transaction Set contains the format and establishes the data contents of the Functional Acknowledgment Transaction Set (997) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to define the control structures for a set of acknowledgments to indicate the results of the syntactical analysis of the electronically encoded documents. The encoded documents are the transaction sets, which are grouped in functional groups, used in defining transactions for business data interchange. This standard does not cover the semantic meaning of the information encoded in the transaction sets.

Table 1 - Header

POS. #	SEG. ID	NAME	REQ. DES.	MAX USE	LOOP REPEAT
0100	ST	Transaction Set Header	M	1	
0200	AK1	Functional Group Response Header	M	1	
		LOOP ID - AK2			>1
0300	AK2	Transaction Set Response Header	O	1	
		LOOP ID - AK2/AK3			>1
0400	AK3	Data Segment Note	O	1	
0500	AK4	Data Element Note	O	99	
0600	AK5	Transaction Set Response Trailer	M	1	
0700	AK9	Functional Group Response Trailer	M	1	
0800	SE	Transaction Set Trailer	M	1	

NOTES:

1/0100 These acknowledgments shall not be acknowledged, thereby preventing an endless cycle of acknowledgments of acknowledgments. Nor shall a Functional Acknowledgment be sent to report errors in a previous Functional Acknowledgment.

1/0100 There is only one Functional Acknowledgment Transaction Set per acknowledged functional group.

1/0100 Only one acknowledgement, either a single Transaction Set 997 or a single Transaction Set 999, should be generated for a functional group unless mutually agreed upon.

1/0200 AK1 is used to respond to the functional group header and to start the acknowledgment for a functional group. There shall be one AK1 segment for the functional group that is being acknowledged.

1/0200 The Functional Acknowledgement is generated at the point of translation, intended for the originator (not any intermediate parties).

1/0200 The Functional Group Header Segment (GS) is used to start the envelope for the Functional Acknowledgment Transaction Sets. In preparing the functional group of acknowledgments, the application sender’s code and the application receiver’s code, taken from the functional group being acknowledged, are exchanged; therefore, one acknowledgment functional group responds to only those functional groups from one application receiver’s code to one application sender’s code.

1/0300 AK2 is used to start the acknowledgment of a transaction set within the received functional group. The AK2 segments shall appear in the same order as the transaction sets in the functional group that has been received and is being acknowledged.

1/0400 The data segments of this standard are used to report the results of the syntactical analysis of the functional groups of transaction sets; they report the extent to which the syntax complies with the standards or proper subsets of transaction sets and functional groups. They do not report on the semantic meaning of the transaction sets (for example, on the ability of the receiver to comply with the request of the sender).

2.4 Segment Detail

This section specifies the segments, data elements, and codes for this implementation. Refer to section 2.1 Presentation Examples for detailed information on the components of the Segment Detail section.

SEGMENT DETAIL

ST - TRANSACTION SET HEADER

X12 Segment Name: Transaction Set Header

X12 Purpose: To indicate the start of a transaction set and to assign a control number

X12 Set Notes:

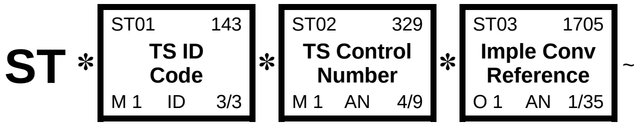
1. These acknowledgments shall not be acknowledged, thereby preventing an endless cycle of acknowledgments of acknowledgments. Nor shall a Functional Acknowledgment be sent to report errors in a previous Functional Acknowledgment.
2. There is only one Functional Acknowledgment Transaction Set per acknowledged functional group.
3. Only one acknowledgement, either a single Transaction Set 997 or a single Transaction Set 999, should be generated for a functional group unless mutually agreed upon.

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **ST✻997✻0001✻005010X230~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**ST01**	**143**	**Transaction Set Identifier Code**	**M 1 ID 3/3**

Code uniquely identifying a Transaction Set

SEMANTIC: The transaction set identifier (ST01) is used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

CODE	DEFINITION
997	**Functional Acknowledgment**

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**ST02**	**329**	**Transaction Set Control Number**	**M 1 AN 4/9**

Identifying control number that must be unique within the transaction set functional group assigned by the originator for a transaction set

The Transaction Set Control Numbers in ST02 and SE02 must be identical. The number is assigned by the originator and must be unique within a functional group (GS-GE). The number also aids in error resolution research. For example, start with the number 0001 and increment from there.

REQUIRED	ST03	1705	**Implementation Convention Reference** Reference assigned to identify Implementation Convention	O 1	AN	1/35

SEMANTIC: The implementation convention reference (ST03) is used by the translation routines of the interchange partners to select the appropriate implementation convention to match the transaction set definition. When used, this implementation convention reference takes precedence over the implementation reference specified in the GS08.

This field contains the same value as data element GS08. This value is always 005010X230. Some translator products strip off the ISA and GS segments prior to application processing. Providing the information from GS08 at this level will help ensure the appropriate application mapping is utilized at translation time.

SEGMENT DETAIL

AK1 - FUNCTIONAL GROUP RESPONSE HEADER

X12 Segment Name: Functional Group Response Header

X12 Purpose: To start acknowledgment of a functional group

X12 Set Notes:

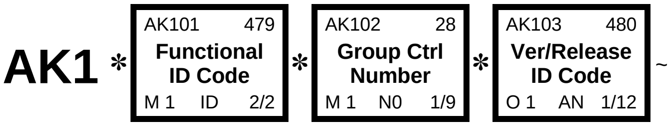
1. AK1 is used to respond to the functional group header and to start the acknowledgment for a functional group. There shall be one AK1 segment for the functional group that is being acknowledged.
2. The Functional Acknowledgement is generated at the point of translation, intended for the originator (not any intermediate parties).
3. The Functional Group Header Segment (GS) is used to start the envelope for the Functional Acknowledgment Transaction Sets. In preparing the functional group of acknowledgments, the application sender's code and the application receiver's code, taken from the functional group being acknowledged, are exchanged; therefore, one acknowledgment functional group responds to only those functional groups from one application receiver's code to one application sender's code.

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **AK1✻0001✻004010~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	AK101	479	**Functional Identifier Code** Code identifying a group of application related transaction sets SEMANTIC: AK101 is the functional ID found in the GS segment (GS01) in the functional group being acknowledged. **Use the value in GS01 from the functional group to which this 997 transaction set is responding.**	M 1	ID	2/2
REQUIRED	AK102	28	**Group Control Number** Assigned number originated and maintained by the sender SEMANTIC: AK102 is the functional group control number found in the GS segment in the functional group being acknowledged. **Use the value in GS06 from the functional group to which this 997 transaction set is responding.**	M 1	N0	1/9

REQUIRED	**AK103**	**480**	**Version / Release / Industry Identifier Code**	**O 1**	**AN**	**1/12**

Code indicating the version, release, subrelease, and industry identifier of the EDI standard being used, including the GS and GE segments; if code in DE455 in GS segment is X, then in DE 480 positions 1-3 are the version number; positions 4-6 are the release and subrelease, level of the version; and positions 7-12 are the industry or trade association identifiers (optionally assigned by user); if code in DE455 in GS segment is T, then other formats are allowed

SEMANTIC: AK103 is the version release industry identifier code in the GS segment (GS08) in the functional group being acknowledged.

CODE SOURCE **881:** Version / Release / Industry Identifier Code

Use the value in GS08 from the functional group to which this 997 transaction set is responding.

SEGMENT DETAIL

AK2 - AK2 TRANSACTION SET RESPONSE

X12 Segment Name: Transaction Set Response Header

X12 Purpose: To start acknowledgment of a single transaction set

X12 Set Notes:

1. AK2 is used to start the acknowledgment of a transaction set within the received functional group. The AK2 segments shall appear in the same order as the transaction sets in the functional group that has been received and is being acknowledged.

Loop: **2000 — AK2 TRANSACTION SET RESPONSE Loop Repeat: 999999**

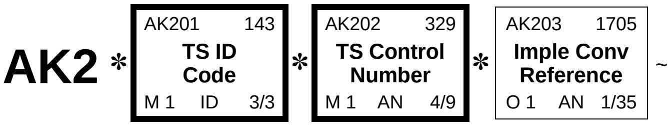
Segment Repeat: **1**

Usage: **SITUATIONAL**

Situational Rule: **Required when an error is present in a transaction set contained in the functional group to which this 997 transaction set is responding. If not required by this implementation guide, may be provided at the sender's discretion but cannot be required by the receiver.**

TR3 Example: **AK2✻0001~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	**AK201**	**143**	**Transaction Set Identifier Code** Code uniquely identifying a Transaction Set SEMANTIC: AK201 is the transaction set ID found in the ST segment (ST01) in the transaction set being acknowledged. **Use the value in ST01 from the transaction set to which this 997 transaction set is responding.**	**M 1**	**ID**	**3/3**
REQUIRED	**AK202**	**329**	**Transaction Set Control Number** Identifying control number that must be unique within the transaction set functional group assigned by the originator for a transaction set SEMANTIC: AK202 is the transaction set control number found in the ST segment in the transaction set being acknowledged. **Use the value in ST02 from the transaction set to which this 997 transaction set is responding.**	**M 1**	**AN**	**4/9**

SITUATIONAL	AK203	1705	**Implementation Convention Reference**	O 1	AN	1/35

Reference assigned to identify Implementation Convention

SEMANTIC: AK203 is the implementation convention reference, if any, found in the ST segment (ST03) in the transaction set being acknowledged.

SITUATIONAL RULE: ***Required when the ST03 value is available in the transaction set to which this 997 transaction set is responding. If not required by this implementation guide, do not send.***

When used, this is the value in ST03 from the transaction set to which this 997 transaction set is responding.

SEGMENT DETAIL

AK3 - AK2/AK3 ERROR IDENTIFICATION

X12 Segment Name: Data Segment Note

X12 Purpose: To report errors in a data segment and identify the location of the data segment

X12 Set Notes: **1.** The data segments of this standard are used to report the results of the syntactical analysis of the functional groups of transaction sets; they report the extent to which the syntax complies with the standards or proper subsets of transaction sets and functional groups. They do not report on the semantic meaning of the transaction sets (for example, on the ability of the receiver to comply with the request of the sender).

Loop: **2100 — AK2/AK3 ERROR IDENTIFICATION Loop Repeat: 999999**

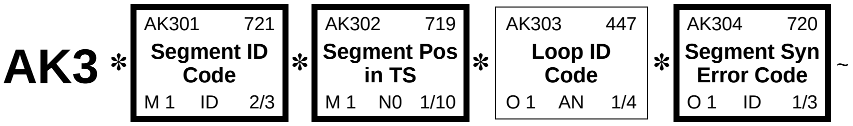
Segment Repeat: **1**

Usage: **SITUATIONAL**

Situational Rule: **Required when an error is present in the transaction set identified in this AK2 loop and the location of the data segment containing the error can be identified by the submitter of this 997. If not required by this implementation guideline, do not send.**

TR3 Example: **AK3✻27✻✻3~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**AK301**	**721**	**Segment ID Code** Code defining the segment ID of the data segment in error (See Appendix A - Number 77) CODE SOURCE **77:** X12 Directories	**M 1 ID 2/3**
REQUIRED	**AK302**	**719**	**Segment Position in Transaction Set** The numerical count position of this data segment from the start of the transaction set: the transaction set header is count position 1	**M 1 N0 1/10**
SITUATIONAL	**AK303**	**447**	**Loop Identifier Code** The loop ID number given on the transaction set diagram is the value for this data element in segments LS and LE SITUATIONAL RULE: ***Required when the data segment containing the error is within a loop and the loop identifier is known by the submitter of this 997. If not required by this implementation guideline, do not send.***	**O 1 AN 1/4**

REQUIRED **AK304** **720** **Segment Syntax Error Code** **O 1** **ID** **1/3**

Code indicating error found based on the syntax editing of a segment

CODE	DEFINITION
1	**Unrecognized segment ID**
2	**Unexpected segment**
3	**Mandatory segment missing**
4	**Loop Occurs Over Maximum Times**
5	**Segment Exceeds Maximum Use**
6	**Segment Not in Defined Transaction Set**
7	**Segment Not in Proper Sequence**
8	**Segment Has Data Element Errors**

SEGMENT DETAIL

AK4 - DATA ELEMENT NOTE

X12 Segment Name: Data Element Note

X12 Purpose: To report errors in a data element or composite data structure and identify the location of the data element

Loop: **2100 — AK2/AK3 ERROR IDENTIFICATION**

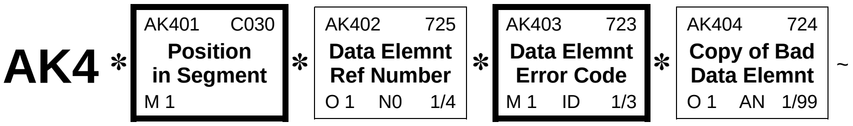
Segment Repeat: **99**

Usage: **SITUATIONAL**

Situational Rule: **Required when the error in the segment described in the AK3 segment applies to a data element and the location of the data element containing the error can be identified by the submitter of the 997. If not required by this implementation guideline, do not send.**

TR3 Example: **AK4✻2✻✻1~**

DIAGRAM



AK4 ✻

AK401 C030	AK402 725	AK403 723	AK404 724
Position in Segment	**Data Elemnt Ref Number**	**Data Elemnt Error Code**	**Copy of Bad Data Elemnt**
M 1	O 1 N0 1/4	M 1 ID 1/3	O 1 AN 1/99

✻ ✻ ✻ ~

ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**AK401**	**C030**	**POSITION IN SEGMENT** Code indicating the relative position of the simple data element or composite data structure in error within a segment, count beginning with 1 for the position immediately following the segment ID; additionally indicating the relative position of a repeating structure in error, count beginning with 1 for the position immediately following the preceding element separator; additionally indicating the relative position of a component of a composite data structure in error, count beginning with 1 for the position following the preceding element or repetition separator	**M 1**
REQUIRED	**AK401 - 1**	**722**	**Element Position in Segment** This is used to indicate the relative position of a simple data element, or the relative position of a composite data structure with the relative position of the component within the composite data structure, in error; in the data segment the count starts with 1 for the simple data element or composite data structure immediately following the segment ID	**M N0 1/2**
SITUATIONAL	**AK401 - 2**	**1528**	**Component Data Element Position in Composite** To identify the component data element position within the composite that is in error SITUATIONAL RULE: ***Required when the error described in this segment relates to a component data element within a composite data structure. If not required, do not send.***	**O N0 1/2**

SITUATIONAL **AK401 - 3** **1686** **Repeating Data Element Position** **O** **N0** **1/4**

To identify the specific repetition of a data element that is in error

SITUATIONAL RULE: ***Required when the error described in this segment relates to a repeating data element. If not required, do not send.***

SITUATIONAL **AK402** **725** **Data Element Reference Number** **O 1** **N0** **1/4**

Reference number used to locate the data element in the Data Element Dictionary

SITUATIONAL RULE: ***Required when the data element reference number for the errored data is known. If not required, do not send.***

CODE SOURCE **77:** X12 Directories

REQUIRED **AK403** **723** **Data Element Syntax Error Code** **M 1** **ID** **1/3**

Code indicating the error found after syntax edits of a data element

CODE	DEFINITION
1	**Mandatory data element missing**
2	**Conditional required data element missing.**
3	**Too many data elements.**
4	**Data element too short.**
5	**Data element too long.**
6	**Invalid character in data element.**
7	**Invalid code value.**
8	**Invalid Date**
9	**Invalid Time**
10	**Exclusion Condition Violated**
12	**Too Many Repetitions**
13	**Too Many Components**

SITUATIONAL **AK404** **724** **Copy of Bad Data Element** **O 1** **AN** **1/99**

This is a copy of the data element in error

SEMANTIC: In no case shall a value be used for AK404 that would generate a syntax error, e.g., an invalid character.

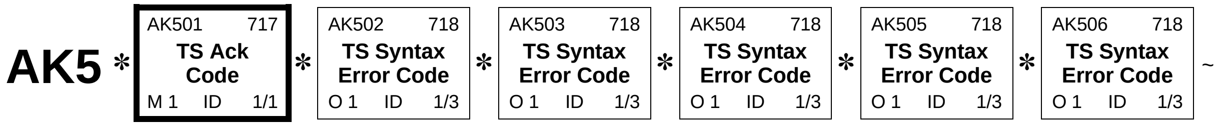
SITUATIONAL RULE: ***Required unless invalid characters are present or data is missing. If not required by this implementation guide, do not send.***

SEGMENT DETAIL

AK5 - TRANSACTION SET RESPONSE

X12 Segment Name:	Transaction Set Response Trailer
X12 Purpose:	To acknowledge acceptance or rejection and report errors in a transaction set
Loop:	**2000 — AK2 TRANSACTION SET RESPONSE**
Segment Repeat:	**1**
Usage:	**REQUIRED**
TR3 Example:	**AK5✻R✻5~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**AK501**	**717**	**Transaction Set Acknowledgment Code** Code indicating accept or reject condition based on the syntax editing of the transaction set	**M 1 ID 1/1**

CODE	DEFINITION
A	**Accepted**
E	**Accepted But Errors Were Noted** **The transaction set indicated in this AK2 loop contained errors, but was forwarded for further processing.**
M	**Rejected, Message Authentication Code (MAC) Failed**
R	**Rejected** **The transaction set indicated in this AK2 loop contained errors, and was NOT forwarded for further processing. It will need to be corrected and resubmitted.**
W	**Rejected, Assurance Failed Validity Tests**
X	**Rejected, Content After Decryption Could Not Be Analyzed**

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
SITUATIONAL	**AK502**	**718**	**Transaction Set Syntax Error Code** Code indicating error found based on the syntax editing of a transaction set	**O 1 ID 1/3**

SITUATIONAL RULE: ***Required when AK501 = E or R. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	**Transaction Set Not Supported**
2	**Transaction Set Trailer Missing**

3	Transaction Set Control Number in Header and Trailer Do Not Match
4	Number of Included Segments Does Not Match Actual Count
5	One or More Segments in Error
6	Missing or Invalid Transaction Set Identifier
7	Missing or Invalid Transaction Set Control Number
8	Authentication Key Name Unknown
9	Encryption Key Name Unknown
10	Requested Service (Authentication or Encrypted) Not Available
11	Unknown Security Recipient
12	Incorrect Message Length (Encryption Only)
13	Message Authentication Code Failed
15	Unknown Security Originator
16	Syntax Error in Decrypted Text
17	Security Not Supported
18	Transaction Set not in Functional Group
19	Invalid Transaction Set Implementation Convention Reference
23	Transaction Set Control Number Not Unique within the Functional Group
24	S3E Security End Segment Missing for S3S Security Start Segment
25	S3S Security Start Segment Missing for S3E Security End Segment
26	S4E Security End Segment Missing for S4S Security Start Segment
27	S4S Security Start Segment Missing for S4E Security End Segment

SITUATIONAL **AK503** **718** **Transaction Set Syntax Error Code** **O 1** **ID** **1/3**

Code indicating error found based on the syntax editing of a transaction set

SITUATIONAL RULE: ***Required when AK501 = E or R, and AK502 has been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	Transaction Set Not Supported
2	Transaction Set Trailer Missing
3	Transaction Set Control Number in Header and Trailer Do Not Match
4	Number of Included Segments Does Not Match Actual Count
5	One or More Segments in Error
6	Missing or Invalid Transaction Set Identifier
7	Missing or Invalid Transaction Set Control Number
8	Authentication Key Name Unknown
9	Encryption Key Name Unknown

10	Requested Service (Authentication or Encrypted) Not Available
11	Unknown Security Recipient
12	Incorrect Message Length (Encryption Only)
13	Message Authentication Code Failed
15	Unknown Security Originator
16	Syntax Error in Decrypted Text
17	Security Not Supported
18	Transaction Set not in Functional Group
19	Invalid Transaction Set Implementation Convention Reference
23	Transaction Set Control Number Not Unique within the Functional Group
24	S3E Security End Segment Missing for S3S Security Start Segment
25	S3S Security Start Segment Missing for S3E Security End Segment
26	S4E Security End Segment Missing for S4S Security Start Segment
27	S4S Security Start Segment Missing for S4E Security End Segment

SITUATIONAL **AK504** **718** **Transaction Set Syntax Error Code** **O 1** **ID** **1/3**

Code indicating error found based on the syntax editing of a transaction set

SITUATIONAL RULE: ***Required when AK501 = E or R, and AK502 and AK503 have been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	Transaction Set Not Supported
2	Transaction Set Trailer Missing
3	Transaction Set Control Number in Header and Trailer Do Not Match
4	Number of Included Segments Does Not Match Actual Count
5	One or More Segments in Error
6	Missing or Invalid Transaction Set Identifier
7	Missing or Invalid Transaction Set Control Number
8	Authentication Key Name Unknown
9	Encryption Key Name Unknown
10	Requested Service (Authentication or Encrypted) Not Available
11	Unknown Security Recipient
12	Incorrect Message Length (Encryption Only)
13	Message Authentication Code Failed
15	Unknown Security Originator
16	Syntax Error in Decrypted Text
17	Security Not Supported
18	Transaction Set not in Functional Group

19	**Invalid Transaction Set Implementation Convention Reference**
23	**Transaction Set Control Number Not Unique within the Functional Group**
24	**S3E Security End Segment Missing for S3S Security Start Segment**
25	**S3S Security Start Segment Missing for S3E Security End Segment**
26	**S4E Security End Segment Missing for S4S Security Start Segment**
27	**S4S Security Start Segment Missing for S4E Security End Segment**

SITUATIONAL **AK505** **718** **Transaction Set Syntax Error Code** **O 1** **ID** **1/3**

Code indicating error found based on the syntax editing of a transaction set

SITUATIONAL RULE: ***Required when AK501 = E or R, and AK502, AK503, and AK504 have been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	**Transaction Set Not Supported**
2	**Transaction Set Trailer Missing**
3	**Transaction Set Control Number in Header and Trailer Do Not Match**
4	**Number of Included Segments Does Not Match Actual Count**
5	**One or More Segments in Error**
6	**Missing or Invalid Transaction Set Identifier**
7	**Missing or Invalid Transaction Set Control Number**
8	**Authentication Key Name Unknown**
9	**Encryption Key Name Unknown**
10	**Requested Service (Authentication or Encrypted) Not Available**
11	**Unknown Security Recipient**
12	**Incorrect Message Length (Encryption Only)**
13	**Message Authentication Code Failed**
15	**Unknown Security Originator**
16	**Syntax Error in Decrypted Text**
17	**Security Not Supported**
18	**Transaction Set not in Functional Group**
19	**Invalid Transaction Set Implementation Convention Reference**
23	**Transaction Set Control Number Not Unique within the Functional Group**
24	**S3E Security End Segment Missing for S3S Security Start Segment**
25	**S3S Security Start Segment Missing for S3E Security End Segment**

26	S4E Security End Segment Missing for S4S Security Start Segment
27	S4S Security Start Segment Missing for S4E Security End Segment

SITUATIONAL	AK506	718	**Transaction Set Syntax Error Code**	O 1	ID	1/3

Code indicating error found based on the syntax editing of a transaction set

SITUATIONAL RULE: ***Required when AK501 = E or R, and AK502, AK503, AK504, and AK505 have been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	Transaction Set Not Supported
2	Transaction Set Trailer Missing
3	Transaction Set Control Number in Header and Trailer Do Not Match
4	Number of Included Segments Does Not Match Actual Count
5	One or More Segments in Error
6	Missing or Invalid Transaction Set Identifier
7	Missing or Invalid Transaction Set Control Number
8	Authentication Key Name Unknown
9	Encryption Key Name Unknown
10	Requested Service (Authentication or Encrypted) Not Available
11	Unknown Security Recipient
12	Incorrect Message Length (Encryption Only)
13	Message Authentication Code Failed
15	Unknown Security Originator
16	Syntax Error in Decrypted Text
17	Security Not Supported
18	Transaction Set not in Functional Group
19	Invalid Transaction Set Implementation Convention Reference
23	Transaction Set Control Number Not Unique within the Functional Group
24	S3E Security End Segment Missing for S3S Security Start Segment
25	S3S Security Start Segment Missing for S3E Security End Segment
26	S4E Security End Segment Missing for S4S Security Start Segment
27	S4S Security Start Segment Missing for S4E Security End Segment

SEGMENT DETAIL

AK9 - FUNCTIONAL GROUP RESPONSE TRAILER

X12 Segment Name: Functional Group Response Trailer

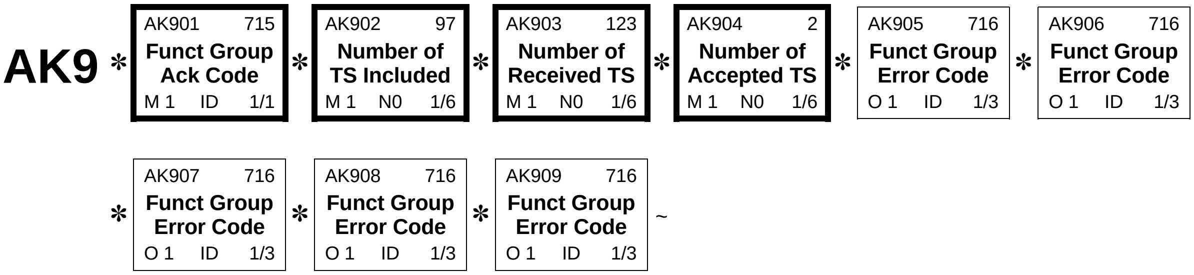
X12 Purpose: To acknowledge acceptance or rejection of a functional group and report the number of included transaction sets from the original trailer, the accepted sets, and the received sets in this functional group

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **AK9✻R✻1✻1✻0~**

DIAGRAM



AK9 ✻

AK901 715 **Funct Group Ack Code** M 1 ID 1/1	✻ AK902 97 **Number of TS Included** M 1 N0 1/6	✻ AK903 123 **Number of Received TS** M 1 N0 1/6	✻ AK904 2 **Number of Accepted TS** M 1 N0 1/6	✻ AK905 716 **Funct Group Error Code** O 1 ID 1/3	✻ AK906 716 **Funct Group Error Code** O 1 ID 1/3
✻ AK907 716 **Funct Group Error Code** O 1 ID 1/3	✻ AK908 716 **Funct Group Error Code** O 1 ID 1/3	✻ AK909 716 **Funct Group Error Code** O 1 ID 1/3	~		

ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**AK901**	**715**	**Functional Group Acknowledge Code**	**M 1 ID 1/1**

Code indicating accept or reject condition based on the syntax editing of the functional group

COMMENT: If AK901 contains the value “A” or “E”, then the transmitted functional group is accepted.

CODE	DEFINITION
A	**Accepted**
E	**Accepted, But Errors Were Noted.**
	The functional group indicated in this 997 transaction set contained errors, but was forwarded for further processing.
M	**Rejected, Message Authentication Code (MAC) Failed**
P	**Partially Accepted, At Least One Transaction Set Was Rejected**
R	**Rejected**
	The functional group indicated in this 997 transaction set contained errors, and was NOT forwarded for further processing. It will need to be corrected and resubmitted.
W	**Rejected, Assurance Failed Validity Tests**

X	**Rejected, Content After Decryption Could Not Be Analyzed**

REQUIRED	**AK902**	**97**	**Number of Transaction Sets Included** Total number of transaction sets included in the functional group or interchange (transmission) group terminated by the trailer containing this data element	**M 1**	**N0**	**1/6**
REQUIRED	**AK903**	**123**	**Number of Received Transaction Sets** Number of Transaction Sets received	**M 1**	**N0**	**1/6**
REQUIRED	**AK904**	**2**	**Number of Accepted Transaction Sets** Number of accepted Transaction Sets in a Functional Group	**M 1**	**N0**	**1/6**
SITUATIONAL	**AK905**	**716**	**Functional Group Syntax Error Code** Code indicating error found based on the syntax editing of the functional group header and/or trailer	**O 1**	**ID**	**1/3**

SITUATIONAL RULE: ***Required when AK901 = E or R, and the error is at the functional group level. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	**Functional Group Not Supported**
2	**Functional Group Version Not Supported**
3	**Functional Group Trailer Missing**
4	**Group Control Number in the Functional Group Header and Trailer Do Not Agree**
5	**Number of Included Transaction Sets Does Not Match Actual Count**
6	**Group Control Number Violates Syntax**
10	**Authentication Key Name Unknown**
11	**Encryption Key Name Unknown**
12	**Requested Service (Authentication or Encryption) Not Available**
13	**Unknown Security Recipient**
14	**Unknown Security Originator**
15	**Syntax Error in Decrypted Text**
16	**Security Not Supported**
17	**Incorrect Message Length (Encryption Only)**
18	**Message Authentication Code Failed**
19	**Functional Group Control Number not Unique within Interchange**
23	**S3E Security End Segment Missing for S3S Security Start Segment**
24	**S3S Security Start Segment Missing for S3E End Segment**
25	**S4E Security End Segment Missing for S4S Security Start Segment**
26	**S4S Security Start Segment Missing for S4E Security End Segment**

SITUATIONAL AK906 716 **Functional Group Syntax Error Code** O 1 ID 1/3

Code indicating error found based on the syntax editing of the functional group header and/or trailer

SITUATIONAL RULE: ***Required when AK901 = E or R, and AK905 has been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	**Functional Group Not Supported**
2	**Functional Group Version Not Supported**
3	**Functional Group Trailer Missing**
4	**Group Control Number in the Functional Group Header and Trailer Do Not Agree**
5	**Number of Included Transaction Sets Does Not Match Actual Count**
6	**Group Control Number Violates Syntax**
10	**Authentication Key Name Unknown**
11	**Encryption Key Name Unknown**
12	**Requested Service (Authentication or Encryption) Not Available**
13	**Unknown Security Recipient**
14	**Unknown Security Originator**
15	**Syntax Error in Decrypted Text**
16	**Security Not Supported**
17	**Incorrect Message Length (Encryption Only)**
18	**Message Authentication Code Failed**
19	**Functional Group Control Number not Unique within Interchange**
23	**S3E Security End Segment Missing for S3S Security Start Segment**
24	**S3S Security Start Segment Missing for S3E End Segment**
25	**S4E Security End Segment Missing for S4S Security Start Segment**
26	**S4S Security Start Segment Missing for S4E Security End Segment**

SITUATIONAL AK907 716 **Functional Group Syntax Error Code** O 1 ID 1/3

Code indicating error found based on the syntax editing of the functional group header and/or trailer

SITUATIONAL RULE: ***Required when AK901 = E or R, and AK905 and AK906 have been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	**Functional Group Not Supported**
2	**Functional Group Version Not Supported**
3	**Functional Group Trailer Missing**
4	**Group Control Number in the Functional Group Header and Trailer Do Not Agree**

5	Number of Included Transaction Sets Does Not Match Actual Count
6	Group Control Number Violates Syntax
10	Authentication Key Name Unknown
11	Encryption Key Name Unknown
12	Requested Service (Authentication or Encryption) Not Available
13	Unknown Security Recipient
14	Unknown Security Originator
15	Syntax Error in Decrypted Text
16	Security Not Supported
17	Incorrect Message Length (Encryption Only)
18	Message Authentication Code Failed
19	Functional Group Control Number not Unique within Interchange
23	S3E Security End Segment Missing for S3S Security Start Segment
24	S3S Security Start Segment Missing for S3E End Segment
25	S4E Security End Segment Missing for S4S Security Start Segment
26	S4S Security Start Segment Missing for S4E Security End Segment

SITUATIONAL	**AK908**	**716**	**Functional Group Syntax Error Code**	**O 1**	**ID**	**1/3**

Code indicating error found based on the syntax editing of the functional group header and/or trailer

SITUATIONAL RULE: ***Required when AK901 = E or R, and AK905, AK906, and AK907 have been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	Functional Group Not Supported
2	Functional Group Version Not Supported
3	Functional Group Trailer Missing
4	Group Control Number in the Functional Group Header and Trailer Do Not Agree
5	Number of Included Transaction Sets Does Not Match Actual Count
6	Group Control Number Violates Syntax
10	Authentication Key Name Unknown
11	Encryption Key Name Unknown
12	Requested Service (Authentication or Encryption) Not Available
13	Unknown Security Recipient
14	Unknown Security Originator
15	Syntax Error in Decrypted Text
16	Security Not Supported
17	Incorrect Message Length (Encryption Only)

18	**Message Authentication Code Failed**
19	**Functional Group Control Number not Unique within Interchange**
23	**S3E Security End Segment Missing for S3S Security Start Segment**
24	**S3S Security Start Segment Missing for S3E End Segment**
25	**S4E Security End Segment Missing for S4S Security Start Segment**
26	**S4S Security Start Segment Missing for S4E Security End Segment**

SITUATIONAL **AK909** **716** **Functional Group Syntax Error Code** **O 1** **ID** **1/3**

Code indicating error found based on the syntax editing of the functional group header and/or trailer

SITUATIONAL RULE: ***Required when AK901 = E or R, and AK905, AK906, AK907, and AK908 have been used, and there are additional error codes to report. If not required by this implementation guide, do not send.***

CODE	DEFINITION
1	**Functional Group Not Supported**
2	**Functional Group Version Not Supported**
3	**Functional Group Trailer Missing**
4	**Group Control Number in the Functional Group Header and Trailer Do Not Agree**
5	**Number of Included Transaction Sets Does Not Match Actual Count**
6	**Group Control Number Violates Syntax**
10	**Authentication Key Name Unknown**
11	**Encryption Key Name Unknown**
12	**Requested Service (Authentication or Encryption) Not Available**
13	**Unknown Security Recipient**
14	**Unknown Security Originator**
15	**Syntax Error in Decrypted Text**
16	**Security Not Supported**
17	**Incorrect Message Length (Encryption Only)**
18	**Message Authentication Code Failed**
19	**Functional Group Control Number not Unique within Interchange**
23	**S3E Security End Segment Missing for S3S Security Start Segment**
24	**S3S Security Start Segment Missing for S3E End Segment**
25	**S4E Security End Segment Missing for S4S Security Start Segment**
26	**S4S Security Start Segment Missing for S4E Security End Segment**

SEGMENT DETAIL

SE - TRANSACTION SET TRAILER

X12 Segment Name: Transaction Set Trailer

X12 Purpose: To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments)

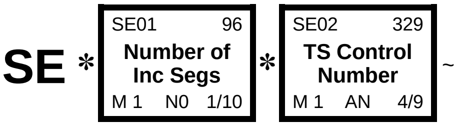
X12 Comments: **1.** SE is the last segment of each transaction set.

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **SE✻53✻0001~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	**SE01**	**96**	**Number of Included Segments** Total number of segments included in a transaction set including ST and SE segments	**M 1**	**N0**	**1/10**
REQUIRED	**SE02**	**329**	**Transaction Set Control Number** Identifying control number that must be unique within the transaction set functional group assigned by the originator for a transaction set	**M 1**	**AN**	**4/9**

3 Example

3.1 EDI Transmission Example

The following example describes a 997 transaction set that is responding to a functional group that was received containing two 837 transaction sets. The first transaction set conformed fully with the X12 standard, while the second contained errors.

The Interchange Control and Functional Group segments (ISA, GS, GE, and IEA) are required in the ASC X12 message. See Appendix C for additional details on these segments.

```
ISA*00*          *00*          *ZZ*123456789
*ZZ*987654321
*041117*1024*U*00501*000000286*0*P*>~
```

```
GS*FA*RCVR*SNDR*20041117*1024*287*X*005010X230~
```

The ST segment indicates the beginning of the 997 transaction set, control number 2870001.

```
ST*997*2870001*005010X230~
```

The AK1 segment describes the functional group to which this 997 is responding.

```
AK1*HC*17456*004010X098A1~
```

The first Transaction Response Loop indicates that the received transaction set, control number 0001, was accepted with no errors.

```
AK2*837*0001~
AK5*A~
```

The second Transaction Response Loop indicates that the received transaction set, control number 0002, was rejected due to a missing CLM01 data element.

```
AK2*837*0002~
AK3*CLM*22**8~
AK4*1*1028*1~
AK5*R*5~
```

The Trailer section provides a summary of the disposition of the received functional group, and ends the transaction set.

```
AK9*P*2*2*1~
SE*11*2870001~
GE*1*287~
IEA*1*000000286~
```

A External Code Sources

77 X12 Directories

SIMPLE DATA ELEMENT/CODE REFERENCES

721, 725

SOURCE

X12.3 Data Element Dictionary
X12.22 Segment Directory

AVAILABLE FROM

Data Interchange Standards Association, Inc. (DISA)
7600 Leesburg Pike
Suite 430
Falls Church, VA 22043

ABSTRACT

The data element dictionary contains the format and descriptions of data elements used to construct X12 segments. It also contains code lists associated with these data elements. The segment directory contains the format and definitions of the data segments used to construct X12 transaction sets.

881 Version / Release / Industry Identifier Code

SIMPLE DATA ELEMENT/CODE REFERENCES

480

SOURCE

Data Interchange Standards Association

AVAILABLE FROM

Data Interchange Standards Association, Inc. (DISA)
7600 Leesburg Pike
Suite 430
Falls Church, VA 22043

ABSTRACT

Code indicating the version, release, sub-release, and industry identifier of the EDI standard being used, including the GS and GE segments; if code in DE455 in GS segment is X, then in DE 480 positions 1-3 are the version number; positions 4-6 are the release and sub-release, level of the version; and positions 7-12 are the industry or trade association identifiers (optionally assigned by user); if code in DE455 in GS segment is T, then other formats are allowed.

B Nomenclature

B.1 ASC X12 Nomenclature

B.1.1 Interchange and Application Control Structures

Appendix B is provided as a reference to the X12 syntax, usage, and related information. It is not a full statement of Interchange and Control Structure rules. The full X12 Interchange and Control Structures and other rules (X12.5, X12.6, X12.59, X12 dictionaries, other X12 standards and official documents) apply unless specifically modified in the detailed instructions of this implementation guide (see Section B.1.1.3.1.2 for an example of such a modification).

B.1.1.1 Interchange Control Structure

The transmission of data proceeds according to very strict format rules to ensure the integrity and maintain the efficiency of the interchange. Each business grouping of data is called a transaction set. For instance, a group of benefit enrollments sent from a sponsor to a payer is considered a transaction set.

Each transaction set contains groups of logically related data in units called segments. For instance, the N4 segment used in the transaction set conveys the city, state, ZIP Code, and other geographic information. A transaction set contains multiple segments, so the addresses of the different parties, for example, can be conveyed from one computer to the other. An analogy would be that the transaction set is like a freight train; the segments are like the train's cars; and each segment can contain several data elements the same as a train car can hold multiple crates.

The sequence of the elements within one segment is specified by the ASC X12 standard as well as the sequence of segments in the transaction set. In a more conventional computing environ-

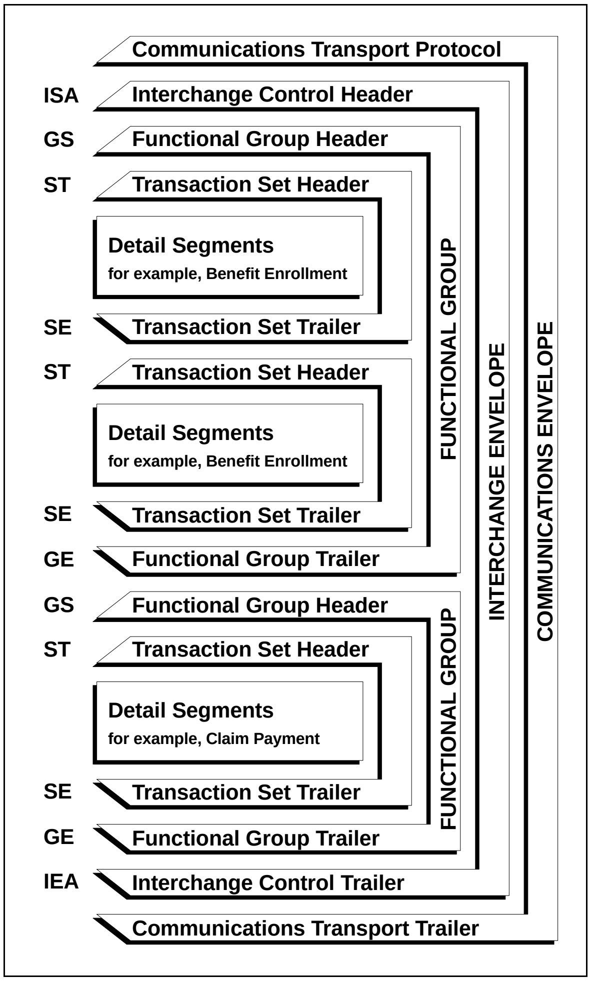


Figure B.1. Transmission Control Schematic

ment, the segments would be equivalent to records, and the elements equivalent to fields.

Similar transaction sets, called “functional groups,” can be sent together within a transmission. Each functional group is prefaced by a group start segment; and a functional group is terminated by a group end segment. One or more functional groups are prefaced by an interchange header and followed by an interchange trailer. Figure B.1., Transmission Control Schematic, illustrates this interchange control.

The interchange header and trailer segments envelop one or more functional groups or interchange-related control segments and perform the following functions:

1. Define the data element separators and the data segment terminator.
2. Identify the sender and receiver.
3. Provide control information for the interchange.
4. Allow for authorization and security information.

B.1.1.2 Application Control Structure Definitions and Concepts

B.1.1.2.1 Basic Structure

A data element corresponds to a data field in data processing terminology. A data segment corresponds to a record in data processing terminology. The data segment begins with a segment ID and contains related data elements. A control segment has the same structure as a data segment; the distinction is in the use. The data segment is used primarily to convey user information, but the control segment is used primarily to convey control information and to group data segments.

B.1.1.2.2 Basic Character Set

The section that follows is designed to have representation in the common character code schemes of EBCDIC, ASCII, and CCITT International Alphabet 5. The ASC X12 standards are graphic-character-oriented; therefore, common character encoding schemes other than those specified herein may be used as long as a common mapping is available. Because the graphic characters have an implied mapping across character code schemes, those bit patterns are not provided here.

The basic character set of this standard, shown in Figure B.2., Basic Character Set, includes those selected from the uppercase letters, digits, space, and special characters as specified below.

A...Z	0...9	!	“	&	’	(	)	*	+
,	-	.	/	:	;	?	=	“ ” (space)	

Figure B.2. Basic Character Set

B.1.1.2.3 Extended Character Set

An extended character set may be used by negotiation between the two parties and includes the lowercase letters and other special characters as specified in Figure B.3., Extended Character Set.

a..z	%	~	@	[	]	_	{
}	\	|	<	>	#	$	

Figure B.3. Extended Character Set

Note that the extended characters include several character codes that have multiple graphical representations for a specific bit pattern. The complete list appears in other standards such as CCITT S.5. Use of the USA graphics for these codes presents no problem unless data is exchanged with an international partner. Other problems, such as the translation of item descriptions from English to French, arise when exchanging data with an international partner, but minimizing the use of codes with multiple graphics eliminates one of the more obvious problems.

For implementations compliant with this guide, either the entire extended character set must be acceptable, or the entire extended character set must not be used. In the absence of a specific trading partner agreement to the contrary, trading partners will assume that the extended character set is acceptable. Use of the extended character set allows the use of the “@” character in email addresses within the PER segment. Users should note that characters in the extended character set, as well as the basic character set, may be used as delimiters only when they do not occur in the data as stated in Section B.1.1.2.7.

B.1.1.2.4 Control Characters

Two control character groups are specified; they have restricted usage. The common notation for these groups is also provided, together with the character coding in three common alphabets. In the Matrix B.1., Base Control Set, the column IA5 represents CCITT V.3 International Alphabet 5.

B.1.1.2.4.1 Base Control Set

The base control set includes those characters that will not have a disruptive effect on most communication protocols. These are represented by:

NOTATION	NAME	EBCDIC	ASCII	IA5
BEL	bell	2F	07	07
HT	horizontal tab	05	09	09
LF	line feed	25	0A	0A
VT	vertical tab	0B	0B	0B
FF	form feed	0C	0C	0C
CR	carriage return	0D	0D	0D
FS	file separator	1C	1C	1C
GS	group separator	1D	1D	1D
RS	record separator	1E	1E	1E
US	unit separator	1F	1F	1F
NL	new line	15		

Matrix B.1. Base Control Set

The Group Separator (GS) may be an exception in this set because it is used in the 3780 communications protocol to indicate blank space compression.

B.1.1.2.4.2 Extended Control Set

The extended control set includes those that may have an effect on a transmission system. These are shown in Matrix B.2., Extended Control Set.

NOTATION	NAME	EBCDIC	ASCII	IA5
SOH	start of header	01	01	01
STX	start of text	02	02	02
ETX	end of text	03	03	03
EOT	end of transmission	37	04	04
ENQ	enquiry	2D	05	05
ACK	acknowledge	2E	06	06
DC1	device control 1	11	11	11
DC2	device control 2	12	12	12
DC3	device control 3	13	13	13
DC4	device control 4	3C	14	14
NAK	negative acknowledge	3D	15	15
SYN	synchronous idle	32	16	16
ETB	end of block	26	17	17

Matrix B.2. Extended Control Set

B.1.1.2.5 Delimiters

A delimiter is a character used to separate two data elements or component elements or to terminate a segment. The delimiters are an integral part of the data.

Delimiters are specified in the interchange header segment, ISA. The ISA segment can be considered in implementations compliant with this guide (see Appendix C, ISA Segment Note 1) to be a 105 byte fixed length record, followed by a segment terminator. The data element separator is byte number 4; the repetition separator is byte number 83; the component element separator is byte number 105; and the segment terminator is the byte that immediately follows the component element separator.

Once specified in the interchange header, the delimiters are not to be used in a data element value elsewhere in the interchange. For consistency, this implementation guide uses the delimiters shown in Matrix B.3., Delimiters, in all examples of EDI transmissions.

CHARACTER	NAME	DELIMITER
*	Asterisk	Data Element Separator
^	Caret	Repetition Separator
:	Colon	Component Element Separator
~	Tilde	Segment Terminator

Matrix B.3. Delimiters

The delimiters above are for illustration purposes only and are not specific recommendations or requirements. Users of this implementation guide should be aware that an application system may use some valid delimiter characters within the application data. Occurrences of delimiter characters in transmitted data within a data element will result in errors in translation. The existence of asterisks (*) within transmitted application data is a known issue that can affect translation software.

B.1.1.3 Business Transaction Structure Definitions and Concepts

The ASC X12 standards define commonly used business transactions (such as a health care claim) in a formal structure called “transaction sets.” A transaction set is composed of a transaction set header control segment, one or more data segments, and a transaction set trailer control segment. Each segment is composed of the following:

- A unique segment ID
- One or more logically related data elements each preceded by a data element separator
- A segment terminator

B.1.1.3.1 Data Element

The data element is the smallest named unit of information in the ASC X12 standard. Data elements are identified as either simple or component. A data element that occurs as an ordinally positioned member of a composite data structure is identified as a component data element. A data element that occurs in a segment outside the defined boundaries of a composite data structure is identified as a simple data element. The distinction between simple and component data elements is strictly a matter of context because a data element can be used in either capacity.

Data elements are assigned a unique reference number. Each data element has a name, description, type, minimum length, and maximum length. For ID type data elements, this guide provides the applicable ASC X12 code values and their descriptions or references where the valid code list can be obtained.

A simple data element within a segment may have an attribute indicating that it may occur once or a specific number of times more than once. The number of permitted repeats are defined as an attribute in the individual segment where the repeated data element occurs.

Each data element is assigned a minimum and maximum length. The length of the data element value is the number of character positions used except as noted for numeric, decimal, and binary elements.

The data element types shown in Matrix B.4., Data Element Types, appear in this implementation guide.

SYMBOL	TYPE
Nn	Numeric
R	Decimal
ID	Identifier
AN	String
DT	Date
TM	Time
B	Binary

Matrix B.4. Data Element Types

The data element minimum and maximum lengths may be restricted in this implementation guide for a compliant implementation. Such restrictions may occur by virtue of the allowed qualifier for the data element or by specific instructions regarding length or format as stated in this implementation guide.

B.1.1.3.1.1 Numeric

A numeric data element is represented by one or more digits with an optional leading sign representing a value in the normal base of 10. The value of a numeric data element includes an implied decimal point. It is used when the position of the decimal point within the data is permanently fixed and is not to be transmitted with the data.

This set of guides denotes the number of implied decimal positions. The representation for this data element type is “Nn” where N indicates that it is numeric and n indicates the number of decimal positions to the right of the implied decimal point.

If n is 0, it need not appear in the specification; N is equivalent to N0. For negative values, the leading minus sign (-) is used. Absence of a sign indicates a positive value. The plus sign (+) must not be transmitted.

EXAMPLE
A transmitted value of 1234, when specified as numeric type N2, represents a value of 12.34.

Leading zeros must be suppressed unless necessary to satisfy a minimum length requirement. The length of a numeric type data element does not include the optional sign.

B.1.1.3.1.2 Decimal

A decimal data element may contain an explicit decimal point and is used for numeric values that have a varying number of decimal positions. This data element type is represented as “R.”

The decimal point always appears in the character stream if the decimal point is at any place other than the right end. If the value is an integer (decimal point at the right end) the decimal point must be omitted. For negative values, the leading minus sign (-) is used. Absence of a sign indicates a positive value. The plus sign (+) must not be transmitted.

Leading zeros must be suppressed unless necessary to satisfy a minimum length requirement. Trailing zeros following the decimal point must be suppressed unless necessary to indicate precision. The use of triad separators (for example, the commas in 1,000,000) is expressly prohibited. The length of a decimal type data element does not include the optional leading sign or decimal point.

EXAMPLE
A transmitted value of 12.34 represents a decimal value of 12.34.

While the ASC X12 standard supports usage of exponential notation, this guide prohibits that usage

For implementation of this guide under the rules promulgated under the Health Insurance Portability and Accountability Act (HIPAA), decimal data elements in Data Element 782 (Monetary Amount) will be limited to a maximum length of 10 characters including reported or implied places for cents (implied value of 00 after the decimal point). Note the statement in the preceding paragraph that the decimal point and leading sign, if sent, are not part of the character count.

EXAMPLE
For implementations mandated under HIPAA rules:

- The following transmitted value represents the largest positive dollar amount that can be sent:
  99999999.99
- The following transmitted value is the longest string of characters that can be sent representing whole dollars.
  99999999
- The following transmitted value is the longest string of characters that can be sent representing negative dollars and cents.
  -99999999.99
- The following transmitted value is the longest string of characters that can be sent representing negative whole dollars.
  -99999999

B.1.1.3.1.3 Identifier

An identifier data element always contains a value from a predefined list of codes that is maintained by the ASC X12 Committee or some other body recognized by the Committee. Trailing spaces must be suppressed unless they are necessary to satisfy a minimum length. An identifier is always left justified. The representation for this data element type is “ID.”

B.1.1.3.1.4 String

A string data element is a sequence of any characters from the basic or extended character sets. The string data element must contain at least one non-space character. The significant characters shall be left justified. Leading spaces, when they occur, are presumed to be significant characters. Trailing spaces must be suppressed unless they are necessary to satisfy a minimum length. The representation for this data element type is “AN.”

B.1.1.3.1.5 Date

A date data element is used to express the standard date in either YYMMDD or CCYYMMDD format in which CC is the first two digits of the calendar year, YY is the last two digits of the calendar year, MM is the month (01 to 12), and DD is the day in the month (01 to 31). The representation for this data element type is “DT.” Users of this guide should note that all dates within transactions are 8-character dates (millennium compliant) in the format CCYYMMDD. The only date data element that is in format YYMMDD is the Interchange Date data element in the ISA segment and the TA1 segment where the century is easily determined because of the nature of an interchange header.

B.1.1.3.1.6 Time

A time data element is used to express the ISO standard time HHMMSSd..d format in which HH is the hour for a 24 hour clock (00 to 23), MM is the minute (00 to 59), SS is the second (00 to 59) and d..d is decimal seconds. The representation for this data element type is “TM.” The length of the data element determines the format of the transmitted time.

EXAMPLE
Transmitted data elements of four characters denote HHMM. Transmitted data elements of six characters denote HHMMSS.

B.1.1.3.2 Repeating Data Elements

Simple or composite data elements within a segment can be designated as repeating data elements. Repeating data elements are adjacent data elements that occur up to a number of times specified in the standard as number of repeats. The implementation guide may also specify the number of repeats of a repeating data element in a specific location in the transaction that are permitted in a compliant implementation. Adjacent occurrences of the same repeating simple data element or composite data structure in a segment shall be separated by a repetition separator.

B.1.1.3.3 Composite Data Structure

The composite data structure is an intermediate unit of information in a segment. Composite data structures are composed of one or more logically related simple data elements, each, except the last, followed by a sub-element separator. The final data element is followed by the next data element separator or the segment terminator. Each simple data element within a composite is called a component.

Each composite data structure has a unique four-character identifier, a name, and a purpose. The identifier serves as a label for the composite. A composite data structure can be further defined through the use of syntax notes, semantic notes, and comments. Each component within the composite is further characterized by a reference designator and a condition designator. The reference designators and the condition designators are described in Sections B.1.1.3.8 and B.1.1.3.9.

A composite data structure within a segment may have an attribute indicating that it may occur once or a specific number of times more than once. The number of permitted repeats are defined as an attribute in the individual segment where the repeated composite data structure occurs.

B.1.1.3.4 Data Segment

The data segment is an intermediate unit of information in a transaction set. In the data stream, a data segment consists of a segment identifier, one or more composite data structures or simple data elements each preceded by a data element separator and succeeded by a segment terminator.

Each data segment has a unique two- or three-character identifier, a name, and a purpose. The identifier serves as a label for the data segment. A segment can be further defined through the use of syntax notes, semantic notes, and comments. Each simple data element or composite data structure within the segment is further characterized by a reference designator and a condition designator.

B.1.1.3.5 Syntax Notes

Syntax notes describe relational conditions among two or more data segment units within the same segment, or among two or more component data elements within the same composite data structure. For a complete description of the relational conditions, See B.1.1.3.9, Condition Designator.

B.1.1.3.6 Semantic Notes

Simple data elements or composite data structures may be referenced by a semantic note within a particular segment. A semantic note provides important additional information regarding the intended meaning of a designated data element, particularly a generic type, in the context of its use within a specific data segment. Semantic notes may also define a relational condition among data elements in a segment based on the presence of a specific value (or one of a set of values) in one of the data elements.

B.1.1.3.7 Comments

A segment comment provides additional information regarding the intended use of the segment.

B.1.1.3.8 Reference Designator

Each simple data element or composite data structure in a segment is provided a structured code that indicates the segment in which it is used and the sequential position within the segment. The code is composed of the segment identifier followed by a two-digit number that defines the position of the simple data element or composite data structure in that segment.

For purposes of creating reference designators, the composite data structure is viewed as the hierarchical equal of the simple data element. Each component data element in a composite data structure is identified by a suffix appended to the reference designator for the composite data structure of which it is a member. This suffix is a two-digit number, prefixed with a hyphen, that defines the position of the component data element in the composite data structure.

EXAMPLE

- The first simple element of the CLP segment would be identified as CLP01.
- The first position in the SVC segment is occupied by a composite data structure that contains seven component data elements, the reference designator for the second component data element would be SVC01-02.

B.1.1.3.9 Condition Designator

This section provides information about X12 standard conditions designators. It is provided so that users will have information about the general standard. Implementation guides may impose other conditions designators. See implementation guide section 2.1 Presentation Examples for detailed information about the implementation guide Industry Usage requirements for compliant implementation.

Data element conditions are of three types: mandatory, optional, and relational. They define the circumstances under which a data element may be required to be present or not present in a particular segment.

DESIGNATOR	DESCRIPTION
M- Mandatory	The designation of mandatory is absolute in the sense that there is no dependency on other data elements. This designation may apply to either simple data elements or composite data structures. If the designation applies to a composite data structure, then at least one value of a component data element in that composite data structure shall be included in the data segment.

DESIGNATOR	DESCRIPTION
O- Optional	The designation of optional means that there is no requirement for a simple data element or composite data structure to be present in the segment. The presence of a value for a simple data element or the presence of value for any of the component data elements of a composite data structure is at the option of the sender.
X- Relational	Relational conditions may exist among two or more simple data elements within the same data segment based on the presence or absence of one of those data elements (presence means a data element must not be empty). Relational conditions are specified by a condition code (see table below) and the reference designators of the affected data elements. A data element may be subject to more than one relational condition. The definitions for each of the condition codes used within syntax notes are detailed below:

CONDITION CODE	DEFINITION
P- Paired or Multiple	If any element specified in the relational condition is present, then all of the elements specified must be present.
R- Required	At least one of the elements specified in the condition must be present.
E- Exclusion	Not more than one of the elements specified in the condition may be present.
C- Conditional	If the first element specified in the condition is present, then all other elements must be present. However, any or all of the elements not specified as the first element in the condition may appear without requiring that the first element be present. The order of the elements in the condition does not have to be the same as the order of the data elements in the data segment.
L- List Conditional	If the first element specified in the condition is present, then at least one of the remaining elements must be present. However, any or all of the elements not specified as the first element in the condition may appear without requiring that the first element be present. The order of the elements in the condition does not have to be the same as the order of the data elements in the data segment.

Table B.5. Condition Designator

B.1.1.3.10 Absence of Data

Any simple data element that is indicated as mandatory must not be empty if the segment is used. At least one component data element of a composite data structure that is indicated as mandatory must not be empty if the segment is used. Optional simple data elements and/or composite data structures and their preceding data element separators that are not needed must be omitted if they occur at the end of a segment. If they do not occur at the end of the segment, the simple data element values and/or composite data structure values may be omitted. Their absence is indicated by the occurrence of their preceding data element separators, in order to maintain the element's or structure's position as defined in the data segment.

Likewise, when additional information is not necessary within a composite, the composite may be terminated by providing the appropriate data element separator or segment terminator.

If a segment has no data in any data element within the segment (an “empty” segment), that segment must not be sent.

B.1.1.3.11 Control Segments

A control segment has the same structure as a data segment, but it is used for transferring control information rather than application information.

B.1.1.3.11.1 Loop Control Segments

Loop control segments are used only to delineate bounded loops. Delineation of the loop shall consist of the loop header (LS segment) and the loop trailer (LE segment). The loop header defines the start of a structure that must contain one or more iterations of a loop of data segments and provides the loop identifier for this loop. The loop trailer defines the end of the structure. The LS segment appears only before the first occurrence of the loop, and the LE segment appears only after the last occurrence of the loop. Unbounded looping structures do not use loop control segments.

B.1.1.3.11.2 Transaction Set Control Segments

The transaction set is delineated by the transaction set header (ST segment) and the transaction set trailer (SE segment). The transaction set header identifies the start and identifier of the transaction set. The transaction set trailer identifies the end of the transaction set and provides a count of the data segments, which includes the ST and SE segments.

B.1.1.3.11.3 Functional Group Control Segments

The functional group is delineated by the functional group header (GS segment) and the functional group trailer (GE segment). The functional group header starts and identifies one or more related transaction sets and provides a control number and application identification information. The functional group trailer defines the end of the functional group of related transaction sets and provides a count of contained transaction sets.

B.1.1.3.11.4 Relations among Control Segments

The control segment of this standard must have a nested relationship as is shown and annotated in this subsection. The letters preceding the control segment name are the segment identifier for that control segment. The indentation of segment identifiers shown below indicates the subordination among control segments.

- **GS** Functional Group Header, starts a group of related transaction sets.
    - **ST** Transaction Set Header, starts a transaction set.
        - **LS** Loop Header, starts a bounded loop of data segments but is not part of the loop.
            - **LS** Loop Header, starts an inner, nested, bounded loop.
            - **LE** Loop Trailer, ends an inner, nested bounded loop.
        - **LE** Loop Trailer, ends a bounded loop of data segments but is not part of the loop.
    - **SE** Transaction Set Trailer, ends a transaction set.
- **GE** Functional Group Trailer, ends a group of related transaction sets.

More than one ST/SE pair, each representing a transaction set, may be used within one functional group. Also more than one LS/LE pair, each representing a bounded loop, may be used within one transaction set.

B.1.1.3.12 Transaction Set

The transaction set is the smallest meaningful set of information exchanged between trading partners. The transaction set consists of a transaction set header segment, one or more data segments in a specified order, and a transaction set trailer segment. See Figure B.1., Transmission Control Schematic.

B.1.1.3.12.1 Transaction Set Header and Trailer

A transaction set identifier uniquely identifies a transaction set. This identifier is the first data element of the Transaction Set Header Segment (ST). A user assigned transaction set control number in the header must match the control number in the Trailer Segment (SE) for any given transaction set. The value for the number of included segments in the SE segment is the total number of segments in the transaction set, including the ST and SE segments.

B.1.1.3.12.2 Data Segment Groups

The data segments in a transaction set may be repeated as individual data segments or as unbounded or bounded loops.

B.1.1.3.12.3 Repeated Occurrences of Single Data Segments

When a single data segment is allowed to be repeated, it may have a specified maximum number of occurrences defined at each specified position within a given transaction set standard. Alternatively, a segment may be allowed to repeat an unlimited number of times. The notation for an unlimited number of repetitions is “>1.”

B.1.1.3.12.4 Loops of Data Segments

Loops are groups of semantically related segments. Data segment loops may be unbounded or bounded.

B.1.1.3.12.4.1 Unbounded Loops

To establish the iteration of a loop, the first data segment in the loop must appear once and only once in each iteration. Loops may have a specified maximum number of repetitions. Alternatively, the loop may be specified as having an unlimited number of iterations. The notation for an unlimited number of repetitions is “>1.”

A specified sequence of segments is in the loop. Loops themselves are optional or mandatory. The requirement designator of the beginning segment of a loop indicates whether at least one occurrence of the loop is required. Each appearance of the beginning segment defines an occurrence of the loop.

The requirement designator of any segment within the loop after the beginning segment applies to that segment for each occurrence of the loop. If there is a mandatory requirement designator for any data segment within the loop after the beginning segment, that data segment is mandatory for each occurrence of the loop. If the loop is optional, the mandatory segment only occurs if the loop occurs.

B.1.1.3.12.4.2 Bounded Loops

The characteristics of unbounded loops described previously also apply to bounded loops. In addition, bounded loops require a Loop Start Segment (LS) to appear before the first occurrence and a Loop End Segment (LE) to appear after the last consecutive occurrence of the loop. If the loop does not occur, the LS and LE segments are suppressed.

B.1.1.3.12.5 Data Segments in a Transaction Set

When data segments are combined to form a transaction set, three characteristics are applied to each data segment: a requirement designator, a position in the transaction set, and a maximum occurrence.

B.1.1.3.12.6 Data Segment Requirement Designators

A data segment, or loop, has one of the following requirement designators for health care and insurance transaction sets, indicating its appearance in the data stream of a transmission. These requirement designators are represented by a single character code.

DESIGNATOR	DESCRIPTION
M- Mandatory	This data segment must be included in the transaction set. (Note that a data segment may be mandatory in a loop of data segments, but the loop itself is optional if the beginning segment of the loop is designated as optional.)
O- Optional	The presence of this data segment is the option of the sending party.

B.1.1.3.12.7 Data Segment Position

The ordinal positions of the segments in a transaction set are explicitly specified for that transaction. Subject to the flexibility provided by the optional requirement designators of the segments, this positioning must be maintained.

B.1.1.3.12.8 Data Segment Occurrence

A data segment may have a maximum occurrence of one, a finite number greater than one, or an unlimited number indicated by “>1.”

B.1.1.3.13 Functional Group

A functional group is a group of similar transaction sets that is bounded by a functional group header segment and a functional group trailer segment. The functional identifier defines the group of transactions that may be included within the functional group. The value for the functional group control number in the header and trailer control segments must be identical for any given group. The value for the number of included transaction sets is the total number of transaction sets in the group. See Figure B.1., Transmission Control Schematic.

B.1.1.4 Envelopes and Control Structures

B.1.1.4.1 Interchange Control Structures

Typically, the term “interchange” connotes the ISA/IEA envelope that is transmitted between trading/business partners. Interchange control is achieved through several “control” components. The interchange control number is contained in data element ISA13 of the ISA segment. The identical control number must also occur in data element 02 of the IEA segment. Most commercial translation software products will verify that these two elements are identical. In most translation software products, if these elements are different the interchange will be “suspended” in error.

There are many other features of the ISA segment that are used for control measures. For instance, the ISA segment contains data elements such as authorization information, security information, sender identification, and receiver identification that can be used for control purposes. These data elements are agreed upon by the trading partners prior to transmission. The interchange date and time data elements as well as the interchange control number within the ISA segment are used for debugging purposes when there is a problem with the transmission or the interchange.

Data Element ISA12, Interchange Control Version Number, indicates the version of the ISA/IEA envelope. GS08 indicates the version of the transaction sets contained within the ISA/IEA envelope. The versions are not required to be the same. An Interchange Acknowledgment can be requested through data element ISA14. The interchange acknowlegement is the TA1 segment. Data element ISA15, Test Indicator, is used between trading partners to indicate that the transmission is in a “test” or “production” mode. Data element ISA16, Subelement Separator, is used by the translator for interpretation of composite data elements.

The ending component of the interchange or ISA/IEA envelope is the IEA segment. Data element IEA01 indicates the number of functional groups that are included within the interchange. In most commercial translation software products, an aggregate count of functional groups is kept while interpreting the interchange. This count is then verified with data element IEA01. If there is a discrepancy, in most commercial products, the interchange is suspended. The other data element in the IEA segment is IEA02 which is referenced above.

See the Appendix C, EDI Control Directory, for a complete detailing of the interchange control header and trailer. The authors recommend that when two transactions with different X12 versions numbers are sent in one interchange control structure (multiple functional groups within one ISA/IEA envelope), the Interchange Control version used should be that of the most recent transaction version included in the envelope. For the transmission of HIPAA transactions with mixed versions, this would be a compliant enveloping structure.

B.1.1.4.2 Functional Groups

Control structures within the functional group envelope include the functional identifier code in GS01. The Functional Identifier Code is used by the commercial translation software during interpretation of the interchange to determine the different transaction sets that may be included within the functional group. If an inappropriate transaction set is contained within the functional group, most commercial translation software will suspend the functional group within the interchange. The Application Sender’s Code in GS02 can be used to identify the sending unit of the transmission. The Application Receiver’s Code in GS03 can be used to identify the receiving unit of the transmission. The functional group contains a creation date (GS04) and creation time (GS05) for the functional group. The Group Control Number is contained in GS06. These data elements (GS04, GS05, and GS06) can be used for debugging purposes. GS08,Version/Release/Industry Identifier Code is the version/release/sub-release of the transaction sets being transmitted in this functional group.

The Functional Group Control Number in GS06 must be identical to data element 02 of the GE segment. Data element GE01 indicates the number of transaction sets within the functional group. In most commercial translation software products, an aggregate count of the transaction sets is kept while interpreting the functional group. This count is then verified with data element GE01.

See the Appendix C, EDI Control Directory, for a complete detailing of the functional group header and trailer.

B.1.1.4.3 HL Structures

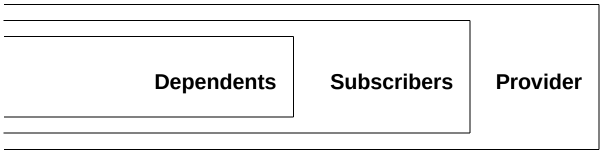
The HL segment is used in several X12 transaction sets to identify levels of detail information using a hierarchical structure, such as relating dependents to a subscriber. Hierarchical levels may differ from guide to guide.

For example, each provider can bill for one or more subscribers, each subscriber can have one or more dependents and the subscriber and the dependents can make one or more claims.

Each guide states what levels are available, the level's usage, number of repeats, and whether that level has subordinate levels within a transaction set.

For implementations compliant with this guide, the repeats of the loops identified by the HL structure shall appear in the hierarchical order specified in BHT01, when those particular hierarchical levels exist. That is, an HL parent loop must be followed by the subordinate child loops, if any, prior to commencing a new HL parent loop at the same hierarchical level.

The following diagram, from transaction set 837, illustrates a typical hierarchy.



The two examples below illustrate this requirement:

Example 1 based on Implementation Guide 811X201:

- INSURER
  - First STATE in transaction (child of INSURER)
    - First POLICY in transaction (child of first STATE)
      - First VEHICLE in transaction (child of first POLICY)
    - Second POLICY in transaction (child of first STATE)
      - Second VEHICLE in transaction (child of second POLICY)
      - Third VEHICLE in transaction (child of second POLICY)
  - Second STATE in transaction (child of INSURER)
    - Third POLICY in transaction (child of second STATE)
      - Fourth VEHICLE in transaction (child of third POLICY)

Example 2 based on Implementation Guide 837X141

- First PROVIDER in transaction
  - First SUBSCRIBER in transaction (child of first PROVIDER)
- Second PROVIDER in transaction
  - Second SUBSCRIBER in transaction (child of second PROVIDER)
    - First DEPENDENT in transaction (child of second SUBSCRIBER)
    - Second DEPENDENT in transaction (child of second SUBSCRIBER)
  - Third SUBSCRIBER in transaction (child of second PROVIDER)
- Third PROVIDER in transaction
  - Fourth SUBSCRIBER in transaction (child of third PROVIDER)
  - Fifth SUBSCRIBER in transaction (child of third PROVIDER
    - Third DEPENDENT in transaction (child of fifth SUBSCRIBER)

B.1.1.5 Acknowledgments

B.1.1.5.1 Interchange Acknowledgment, TA1

The TA1 segment provides the capability for the interchange receiver to notify the sender that a valid envelope was received or that problems were encountered with the interchange control structure. The TA1 verifies the envelopes only. Transaction set-specific verification is accomplished through use of the Functional Acknowledgment Transaction Set, 997. See B.1.1.5.2, Functional Acknowledgment, 997, for more details. The TA1 is unique in that it is a single segment transmitted without the GS/GE envelope structure. A TA1 can be included in an interchange with other functional groups and transactions.

Encompassed in the TA1 are the interchange control number, interchange date and time, interchange acknowledgment code, and the interchange note code. The interchange control number, interchange date and time are identical to those that were present in the transmitted interchange from the trading partner. This provides the capability to associate the TA1 with the transmitted interchange. TA104, Interchange Acknowledgment Code, indicates the status of the interchange control structure. This data element stipulates whether the transmitted interchange was accepted with no errors, accepted with errors, or rejected because of errors. TA105, Interchange Note Code, is a numerical code that indicates the error found while processing the interchange control structure. Values for this data element indicate whether the error occurred at the interchange or functional group envelope.

See the Appendix C, EDI Control Directory, for a complete detailing of the TA1 segment.

B.1.1.5.2 Functional Acknowledgment, 997

The 997 informs the submitter that the functional group arrived at the destination. It may include information about the syntactical quality of the functional group.

The Functional Acknowledgment (997) transaction is not required as a response to receipt of a batch transaction compliant with this implementation guide.

The Functional Acknowledgment (997) transaction is not required as a response to receipt of a real-time transaction compliant with this implementation guide.

B.2 Object Descriptors

Object Descriptors (OD) do not apply to this implementation guide.

C EDI Control Directory Including Implementation Usage

C.1 Control Segments

- **ISA**
  Interchange Control Header Segment
- **TA1**
  Interchange Acknowledgment Header Segment
- **GS**
  Functional Group Header Segment
- **GE**
  Functional Group Trailer Segment
- **IEA**
  Interchange Control Trailer Segment

SEGMENT DETAIL

ISA - INTERCHANGE CONTROL HEADER

X12 Segment Name: Interchange Control Header

X12 Purpose: To start and identify an interchange of zero or more functional groups and interchange-related control segments

Segment Repeat: **1**

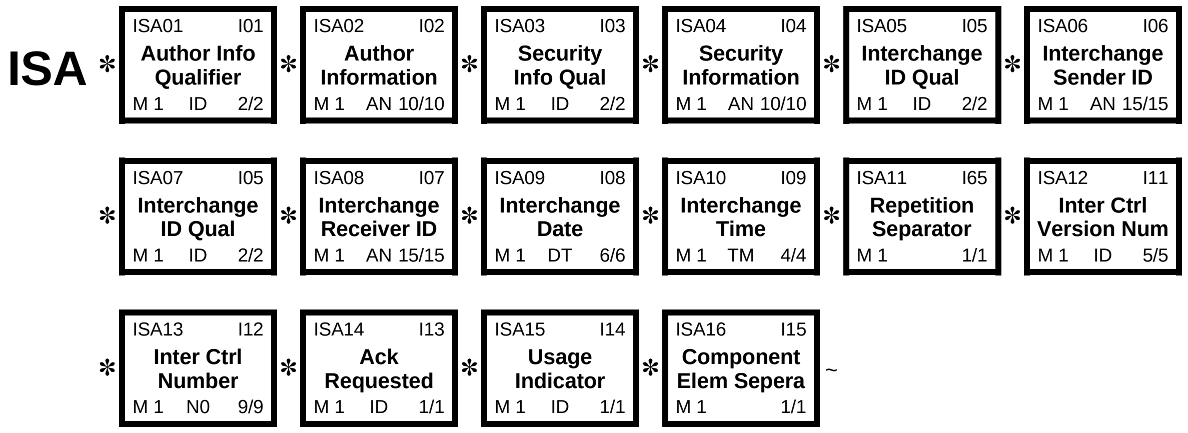
Usage: **REQUIRED**

TR3 Notes:

1. **All positions within each of the data elements must be filled.**
2. **For compliant implementations under this implementation guide, ISA13, the interchange Control Number, must be a positive unsigned number. Therefore, the ISA segment can be considered a fixed record length segment.**
3. **The first element separator defines the element separator to be used through the entire interchange.**
4. **The ISA segment terminator defines the segment terminator used throughout the entire interchange.**
5. **Spaces in the example interchanges are represented by “.” for clarity.**

TR3 Example: **ISA✻00✻..........✻01✻SECRET....✻ZZ✻SUBMITTERS.ID..✻ZZ✻ RECEIVERS.ID...✻030101✻1253✻^✻00501✻000000905✻1✻T✻:~**

DIAGRAM



ISA ✻

ISA01	I01	**Author Info Qualifier**	M 1	ID	2/2
ISA02	I02	**Author Information**	M 1	AN	10/10
ISA03	I03	**Security Info Qual**	M 1	ID	2/2
ISA04	I04	**Security Information**	M 1	AN	10/10
ISA05	I05	**Interchange ID Qual**	M 1	ID	2/2
ISA06	I06	**Interchange Sender ID**	M 1	AN	15/15
ISA07	I05	**Interchange ID Qual**	M 1	ID	2/2
ISA08	I07	**Interchange Receiver ID**	M 1	AN	15/15
ISA09	I08	**Interchange Date**	M 1	DT	6/6
ISA10	I09	**Interchange Time**	M 1	TM	4/4
ISA11	I65	**Repetition Separator**	M 1		1/1
ISA12	I11	**Inter Ctrl Version Num**	M 1	ID	5/5
ISA13	I12	**Inter Ctrl Number**	M 1	N0	9/9
ISA14	I13	**Ack Requested**	M 1	ID	1/1
ISA15	I14	**Usage Indicator**	M 1	ID	1/1
ISA16	I15	**Component Elem Sepera**	M 1		1/1

~

ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	**ISA01**	**I01**	**Authorization Information Qualifier** Code identifying the type of information in the Authorization Information	**M 1**	**ID**	**2/2**
REQUIRED	**ISA02**	**I02**	**Authorization Information** Information used for additional identification or authorization of the interchange sender or the data in the interchange; the type of information is set by the Authorization Information Qualifier (I01)	**M 1**	**AN**	**10/10**
REQUIRED	**ISA03**	**I03**	**Security Information Qualifier** Code identifying the type of information in the Security Information	**M 1**	**ID**	**2/2**
REQUIRED	**ISA04**	**I04**	**Security Information** This is used for identifying the security information about the interchange sender or the data in the interchange; the type of information is set by the Security Information Qualifier (I03)	**M 1**	**AN**	**10/10**
REQUIRED	**ISA05**	**I05**	**Interchange ID Qualifier** Code indicating the system/method of code structure used to designate the sender or receiver ID element being qualified	**M 1**	**ID**	**2/2**
			This ID qualifies the Sender in ISA06.			
REQUIRED	**ISA06**	**I06**	**Interchange Sender ID** Identification code published by the sender for other parties to use as the receiver ID to route data to them; the sender always codes this value in the sender ID element	**M 1**	**AN**	**15/15**
REQUIRED	**ISA07**	**I05**	**Interchange ID Qualifier** Code indicating the system/method of code structure used to designate the sender or receiver ID element being qualified	**M 1**	**ID**	**2/2**
			This ID qualifies the Receiver in ISA08.			
REQUIRED	**ISA08**	**I07**	**Interchange Receiver ID** Identification code published by the receiver of the data; When sending, it is used by the sender as their sending ID, thus other parties sending to them will use this as a receiving ID to route data to them	**M 1**	**AN**	**15/15**
REQUIRED	**ISA09**	**I08**	**Interchange Date** Date of the interchange	**M 1**	**DT**	**6/6**
			The date format is YYMMDD.			
REQUIRED	**ISA10**	**I09**	**Interchange Time** Time of the interchange	**M 1**	**TM**	**4/4**
			The time format is HHMM.			
REQUIRED	**ISA11**	**I65**	**Repetition Separator** Type is not applicable; the repetition separator is a delimiter and not a data element; this field provides the delimiter used to separate repeated occurrences of a simple data element or a composite data structure; this value must be different than the data element separator, component element separator, and the segment terminator	**M 1**		**1/1**
REQUIRED	**ISA12**	**I11**	**Interchange Control Version Number** Code specifying the version number of the interchange control segments	**M 1**	**ID**	**5/5**

CODE	DEFINITION
00501	**Standards Approved for Publication by ASC X12 Procedures Review Board through October 2003**

REQUIRED **ISA13** **I12** **Interchange Control Number** **M 1** **N0** **9/9**

A control number assigned by the interchange sender

The Interchange Control Number, ISA13, must be identical to the associated Interchange Trailer IEA02.

Must be a positive unsigned number and must be identical to the value in IEA02.

REQUIRED **ISA14** **I13** **Acknowledgment Requested** **M 1** **ID** **1/1**

Code indicating sender’s request for an interchange acknowledgment

See Section B.1.1.5.1 for interchange acknowledgment information.

CODE	DEFINITION
0	**No Interchange Acknowledgment Requested**
1	**Interchange Acknowledgment Requested (TA1)**

REQUIRED **ISA15** **I14** **Interchange Usage Indicator** **M 1** **ID** **1/1**

Code indicating whether data enclosed by this interchange envelope is test, production or information

REQUIRED **ISA16** **I15** **Component Element Separator** **M 1** **1/1**

Type is not applicable; the component element separator is a delimiter and not a data element; this field provides the delimiter used to separate component data elements within a composite data structure; this value must be different than the data element separator and the segment terminator

SEGMENT DETAIL

TA1 - INTERCHANGE ACKNOWLEDGMENT

X12 Segment Name: Interchange Acknowledgment

X12 Purpose: To report the status of processing a received interchange header and trailer or the non-delivery by a network provider

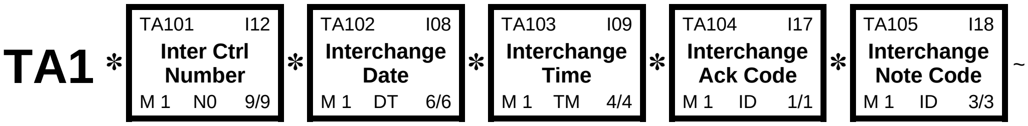
Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Notes:

1. **When requested by the sender (as indicated in the ISA14 of the submitted interchange), or when an interchange is rejected, a TA1 segment may be included in the same interchange as the 997 transaction set. When this is done, the TA1 segment must be included within the interchange (ISA/IEA) and outside of any included functional groups (GS/GE). It is recommended that the TA1 segment be placed between the ISA and GS segments. However, at the sender's discretion, the TA1 segment may be sent within its own interchange (i.e. ISA-TA1-IEA).**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	**TA101**	**I12**	**Interchange Control Number** A control number assigned by the interchange sender **This is the value in ISA13 from the interchange to which this TA1 is responding.**	**M 1**	**N0**	**9/9**
REQUIRED	**TA102**	**I08**	**Interchange Date** Date of the interchange	**M 1**	**DT**	**6/6**
REQUIRED	**TA103**	**I09**	**Interchange Time** Time of the interchange	**M 1**	**TM**	**4/4**
REQUIRED	**TA104**	**I17**	**Interchange Acknowledgment Code** Code indicating the status of the receipt of the interchange control structure	**M 1**	**ID**	**1/1**
REQUIRED	**TA105**	**I18**	**Interchange Note Code** Code specifying the error found processing the interchange control structure	**M 1**	**ID**	**3/3**

SEGMENT DETAIL

GS - FUNCTIONAL GROUP HEADER

X12 Segment Name: Functional Group Header

X12 Purpose: To indicate the beginning of a functional group and to provide control information

X12 Comments:

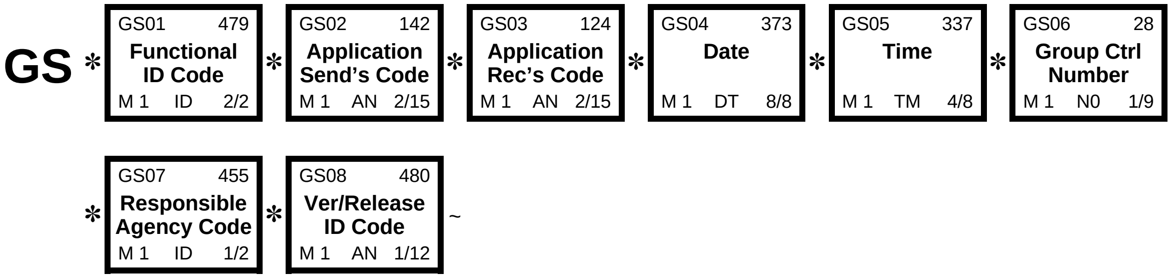
1. A functional group of related transaction sets, within the scope of X12 standards, consists of a collection of similar transaction sets enclosed by a functional group header and a functional group trailer.

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **GS✻XX✻SENDER CODE✻RECEIVER CODE✻19991231✻0802✻1✻X✻005010X230~**

DIAGRAM


GS ✻
GS01 479 Functional ID Code M 1 ID 2/2
✻
GS02 142 Application Send's Code M 1 AN 2/15
✻
GS03 124 Application Rec's Code M 1 AN 2/15
✻
GS04 373 Date M 1 DT 8/8
✻
GS05 337 Time M 1 TM 4/8
✻
GS06 28 Group Ctrl Number M 1 N0 1/9
✻
GS07 455 Responsible Agency Code M 1 ID 1/2
✻
GS08 480 Ver/Release ID Code M 1 AN 1/12
~

ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES
REQUIRED	**GS01**	**479**	**Functional Identifier Code** Code identifying a group of application related transaction sets	**M 1 ID 2/2**
			This is the 2-character Functional Identifier Code assigned to each transaction set by X12. The specific code for a transaction set defined by this implementation guide is presented in section 1.2, Version Information.	
			The functional identifier code required when creating a 997 transaction set is FA.	
REQUIRED	**GS02**	**142**	**Application Sender's Code** Code identifying party sending transmission; codes agreed to by trading partners	**M 1 AN 2/15**
			Use this code to identify the unit sending the information.	
REQUIRED	**GS03**	**124**	**Application Receiver's Code** Code identifying party receiving transmission; codes agreed to by trading partners	**M 1 AN 2/15**
			Use this code to identify the unit receiving the information.	

REQUIRED **GS04** **373** **Date** **M 1 DT 8/8**

Date expressed as CCYYMMDD where CC represents the first two digits of the calendar year

SEMANTIC: GS04 is the group date.

Use this date for the functional group creation date.

REQUIRED **GS05** **337** **Time** **M 1 TM 4/8**

Time expressed in 24-hour clock time as follows: HHMM, or HHMMSS, or HHMMSSD, or HHMMSSDD, where H = hours (00-23), M = minutes (00-59), S = integer seconds (00-59) and DD = decimal seconds; decimal seconds are expressed as follows: D = tenths (0-9) and DD = hundredths (00-99)

SEMANTIC: GS05 is the group time.

Use this time for the creation time. The recommended format is HHMM.

REQUIRED **GS06** **28** **Group Control Number** **M 1 N0 1/9**

Assigned number originated and maintained by the sender

SEMANTIC: The data interchange control number GS06 in this header must be identical to the same data element in the associated functional group trailer, GE02.

In order to provide sufficient discrimination for the acknowledgment process to operate reliably and to ensure that audit trails are unambiguous, the combination of Functional ID Code (GS01), Application Sender’s ID (GS02), Application Receiver’s ID (GS03), and Functional Group Control Numbers (GS06, GE02) shall by themselves be unique within a reasonably extended time frame whose boundaries shall be defined by trading partner agreement. Because at some point it may be necessary to reuse a sequence of control numbers, the Functional Group Date and Time may serve as an additional discriminant only to differentiate functional group identity over the longest possible time frame.

REQUIRED **GS07** **455** **Responsible Agency Code** **M 1 ID 1/2**

Code identifying the issuer of the standard; this code is used in conjunction with Data Element 480

REQUIRED **GS08** **480** **Version / Release / Industry Identifier Code** **M 1 AN 1/12**

Code indicating the version, release, subrelease, and industry identifier of the EDI standard being used, including the GS and GE segments; if code in DE455 in GS segment is X, then in DE 480 positions 1-3 are the version number; positions 4-6 are the release and subrelease, level of the version; and positions 7-12 are the industry or trade association identifiers (optionally assigned by user); if code in DE455 in GS segment is T, then other formats are allowed

CODE SOURCE **881:** Version / Release / Industry Identifier Code

This is the unique Version/Release/Industry Identifier Code assigned to an implementation by X12. The specific code for a transaction set defined by this implementation guide is presented in section 1.2, Version Information.

Functional Acknowledgment (997) transactions must use only the Version/Release of the underlying standard used to create the 997.

CODE	DEFINITION
005010X230	**Standards Approved for Publication by ASC X12 Procedures Review Board through October 2003**

SEGMENT DETAIL

GE - FUNCTIONAL GROUP TRAILER

X12 Segment Name: Functional Group Trailer

X12 Purpose: To indicate the end of a functional group and to provide control information

X12 Comments:

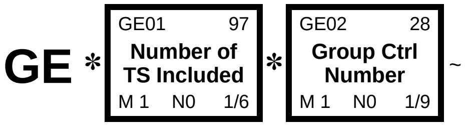
1. The use of identical data interchange control numbers in the associated functional group header and trailer is designed to maximize functional group integrity. The control number is the same as that used in the corresponding header.

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **GE✻1✻1~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	**GE01**	**97**	**Number of Transaction Sets Included** Total number of transaction sets included in the functional group or interchange (transmission) group terminated by the trailer containing this data element	**M 1**	**N0**	**1/6**
REQUIRED	**GE02**	**28**	**Group Control Number** Assigned number originated and maintained by the sender SEMANTIC: The data interchange control number GE02 in this trailer must be identical to the same data element in the associated functional group header, GS06. **The value in GE02 must be identical to the value in GS06.**	**M 1**	**N0**	**1/9**

SEGMENT DETAIL

IEA - INTERCHANGE CONTROL TRAILER

X12 Segment Name: Interchange Control Trailer

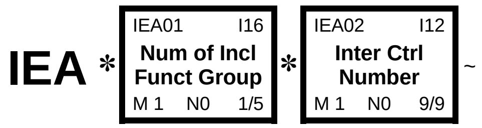
X12 Purpose: To define the end of an interchange of zero or more functional groups and interchange-related control segments

Segment Repeat: **1**

Usage: **REQUIRED**

TR3 Example: **IEA✻1✻000000905~**

DIAGRAM



ELEMENT DETAIL

USAGE	REF. DES.	DATA ELEMENT	NAME	ATTRIBUTES		
REQUIRED	**IEA01**	**I16**	**Number of Included Functional Groups** A count of the number of functional groups included in an interchange	**M 1**	**N0**	**1/5**
REQUIRED	**IEA02**	**I12**	**Interchange Control Number** A control number assigned by the interchange sender **The value in IEA02 must be identical to the value in ISA13.**	**M 1**	**N0**	**9/9**

D Change Summary

This is the first ASC X12N Implementation Guide for the Functional Acknowledgment business use of the 997. In future guides, this section will contain a summary and detail of all changes since the previous guide.